CHAPTER 8

CONSTRAINED OPTIMIZATION 2: SEQUENTIAL QUADRATIC PROGRAMMING, INTERIOR POINT AND GENERALIZED REDUCED GRADIENT METHODS

8.1 Introduction

In the previous chapter we examined the necessary and sufficient conditions for a constrained optimum. We did not, however, discuss any algorithms for constrained optimization. That is the purpose of this chapter.

The three algorithms we will study are three of the most common. Sequential Quadratic Programming (SQP) is a very popular algorithm because of its fast convergence properties. It is available in MATLAB and is widely used. The Interior Point (IP) algorithm has grown in popularity the past 15 years and recently became the default algorithm in MATLAB. It is particularly useful for solving large-scale problems. The Generalized Reduced Gradient method (GRG) has been shown to be effective on highly nonlinear engineering problems and is the algorithm used in Excel.

SQP and IP share a common background. Both of these algorithms apply the Newton-Raphson (NR) technique for solving nonlinear equations to the KKT equations for a modified version of the problem. Thus we will begin with a review of the NR method.

8.2 The Newton-Raphson Method for Solving Nonlinear Equations

Before we get to the algorithms, there is some background we need to cover first. This includes reviewing the Newton-Raphson (NR) method for solving sets of nonlinear equations.

8.2.1 One equation with One Unknown

The NR method is used to find the solution to sets of nonlinear equations. For example, suppose we wish to find the solution to the equation:

$$x + 2 = e^x$$

We cannot solve for x directly. The NR method solves the equation in an iterative fashion based on results derived from the Taylor expansion.

First, the equation is rewritten in the form,

$$x + 2 - e^x = 0 \tag{8.1}$$

We then provide a starting estimate of the value of x that solves the equation. This point becomes the point of expansion for a Taylor series:

$$f_{NR} \approx f_{NR}^0 + \frac{df_{NR}^0}{dx}\left(x - x^0\right) \tag{8.2}$$

where we use the subscript "NR" to distinguish between a function where we are applying the NR method and the objective function of an optimization problem. We would like to drive the value of the function to zero:

$$0 = f_{NR}^0 + \frac{df_{NR}^0}{dx}\left(x - x^0\right) \tag{8.3}$$

If we denote $\Delta x^0 = x - x^0$, and solve for Δx in (8.3):

$$\Delta x^0 = \frac{-f_{NR}^0}{df_{NR}^0 / dx} \tag{8.4}$$

We then add Δx to x^0 to obtain a new guess for the value of x that satisfies the equation, obtain the derivative there, get the new function value, and iterate until the function value or *residual*, goes to zero. The process is illustrated in Fig. 8.1 for the example given in (8.1) with a starting guess $x = 2.0$.

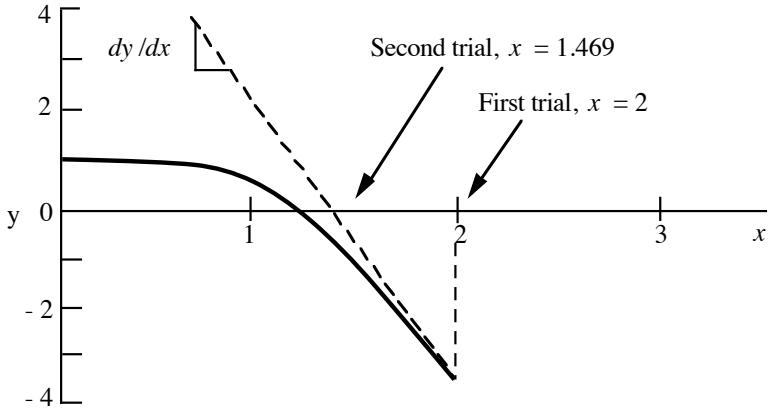

Fig. 8.1 Newton-Raphson method on (8.1)

Numerical results are:

k	x	f(x)	df/dx
1	2	–3.389056	–6.389056
2	1.469553	–0.877738	–3.347291
3	1.20732948	–0.13721157	–2.34454106
4	1.14880563	–0.00561748	–2.15442311
5	1.146198212	–0.000010714	–2.146208926
6	1.1461932206	–0.00000000004	–2.1461932254

For simple roots, NR has *second order convergence*. This means that the number of significant figures in the solution roughly doubles at each iteration. We can see this in the above table where the value of x at iteration 2 has one significant figure (1); at iteration 3 it has one (1); at iteration 4 it has three (1.14); at iteration 5 it has six (1.14619), and so on. We also see that the error in the residual, as indicated by the number of zeros after the decimal point, also decreases in this fashion, i.e., the number of zeros roughly doubles at each iteration.

8.2.2 Multiple Equations with Multiple Unknowns

The NR method is easily extended to solve n equation in n unknowns. Writing the Taylor series for the equations in vector form:

$$
\begin{aligned}
0 &= f_{1NR}^0 + \left(\nabla f_{1NR}^0\right)^{\mathrm{T}} \Delta\mathbf{x} \\
0 &= f_{2NR}^0 + \left(\nabla f_{2NR}^0\right)^{\mathrm{T}} \Delta\mathbf{x} \\
&\cdots \quad \cdots \\
0 &= f_{nNR}^0 + \left(\nabla f_{nNR}^0\right)^{\mathrm{T}} \Delta\mathbf{x}
\end{aligned}
$$

We can rewrite these relationships as:

$$
\begin{bmatrix}
\left(\nabla f_{1NR}^0\right)^{\mathrm{T}} \\
\left(\nabla f_{2NR}^0\right)^{\mathrm{T}} \\
\cdots \\
\left(\nabla f_{nNR}^0\right)^{\mathrm{T}}
\end{bmatrix} \Delta\mathbf{x} =
\begin{bmatrix}
-f_{1NR}^0 \\
-f_{2NR}^0 \\
\cdots \\
-f_{nNR}^0
\end{bmatrix} \tag{8.5}
$$

For 2 X 2 System,

$$
\begin{bmatrix}
\dfrac{\partial f_{1NR}}{\partial x_1} & \dfrac{\partial f_{1NR}}{\partial x_2} \\
\dfrac{\partial f_{2NR}}{\partial x_1} & \dfrac{\partial f_{2NR}}{\partial x_2}
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2
\end{bmatrix} =
\begin{bmatrix}
-f_{1NR} \\
-f_{2NR}
\end{bmatrix} \tag{8.6}
$$

In (8.5) we will denote the vector of residuals at the starting point as $\mathbf{f}_{NR}^0$. We will denote the matrix of coefficients as **G.** Equation (8.5) can then be written,

$$
\mathbf{G}\Delta\mathbf{x}^0 = -\mathbf{f}_{NR}^0 \tag{8.7}
$$

The solution is obviously

$$\Delta \mathbf{x}^0 = -\left(\mathbf{G}^{-1}\right)\mathbf{f}_{NR}^0 \tag{8.8}$$

8.2.3 Using the NR method to Solve the Necessary Conditions

In this section we will make a very important connection—we will apply NR to solve the necessary conditions. Consider for example, a very simple case—an unconstrained problem in two variables with objective function, *f*. We know the necessary conditions are,

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 0 \\ \frac{\partial f}{\partial x_2} &= 0 \end{aligned} \tag{8.9}$$

Now suppose we wish to solve these equations using NR, that is we wish to find the value **x*** that drives the partial derivatives to zero. In terms of notation and discussion this gets a little tricky because the NR method involves taking derivatives of the equations to be solved, and the equations we wish to solve are composed of derivatives. So when we substitute (8.9) into the NR method, we end up with *second* derivatives.

For example, if we set $f_{1NR} = \dfrac{\partial f}{\partial x_1}$ and $f_{2NR} = \dfrac{\partial f}{\partial x_2}$, then we can write (8.6) as,

$$\begin{bmatrix} \dfrac{\partial}{\partial x_1}\left(\dfrac{\partial f}{\partial x_1}\right) & \dfrac{\partial}{\partial x_2}\left(\dfrac{\partial f}{\partial x_1}\right) \\ \dfrac{\partial}{\partial x_1}\left(\dfrac{\partial f}{\partial x_2}\right) & \dfrac{\partial}{\partial x_2}\left(\dfrac{\partial f}{\partial x_2}\right) \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -\left(\dfrac{\partial f}{\partial x_1}\right) \\ -\left(\dfrac{\partial f}{\partial x_2}\right) \end{bmatrix} \tag{8.10}$$

or,

$$\begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_2 \partial x_1} \\ \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_2^2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -\dfrac{\partial f}{\partial x_1} \\ -\dfrac{\partial f}{\partial x_2} \end{bmatrix} \tag{8.11}$$

which should be familiar from Chapter 3, because (8.11) can be written in vector form as,

$$\mathbf{H}\Delta \mathbf{x} = -\nabla f \tag{8.12}$$

and the solution is,

$$\Delta \mathbf{x} = -\left(\mathbf{H}^{-1}\right)\nabla f \tag{8.13}$$

We recognize (8.13) as Newton's method for solving for an unconstrained optimum. Thus we have the important result that *Newton's method is the same as applying NR on the necessary conditions for an unconstrained problem*. From the properties of the NR method, we know that if Newton's method converges (and recall that it doesn't always converge), it will do so with second order convergence.

Both SQP and IP methods extend these ideas to constrained problems.

8.3 The Sequential Quadratic Programming (SQP) Algorithm

8.3.1 Introduction and Problem Definition

The SQP algorithm was developed in the early 1980's primarily by M. J. D. Powell, a mathematician at Cambridge University [23, 24]. SQP works by solving for where the KKT equations are satisfied. SQP is a very efficient algorithm in terms of the number of function calls needed to get to the optimum. It converges to the optimum by simultaneously improving the objective and tightening feasibility of the constraints. Only the optimal design is guaranteed to be feasible; intermediate designs may be infeasible. It requires that we have some means of estimating the active constraints at each step of the algorithm.

We will start with a problem which only has equality constraints. We recall that when we only have equality constraints, we do not have to worry about complementary slackness, which makes thing simpler. So the problem we will focus on at this point is,

$$\text{Min} \quad f(\mathbf{x}) \tag{8.14}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) = 0 \quad i - 1,...,m \tag{8.15}$$

8.3.2 The SQP Approximation

As we have previously mentioned in Chapter 7, a problem with a quadratic objective and linear constraints is known as a *quadratic programming problem*. These problems have a special name because the KKT equations are linear (except for complementary slackness) and are easily solved. We will make a quadratic programming approximation at the point $\mathbf{x}^k$ to the problem (8.14)-(8.15) given by,

$$f_a = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^{\mathrm{T}}\nabla_x^2 \mathrm{L}^k \Delta\mathbf{x} \tag{8.16}$$

$$g_{i,a} = g_i^k + \left(\nabla g_i^k\right)^{\mathrm{T}} \Delta\mathbf{x} = 0 \quad i = 1,\ldots,m \tag{8.17}$$

where the subscript *a* indicates the approximation. Close examination of (8.16) shows something unexpected. Instead of $\nabla^2 f$ as we would normally have if we were doing a Taylor approximation of the objective, we have $\nabla_x^2 L$, the Hessian of the Lagrangian function with respect to x. Why is this the case? It is directly tied to applying NR on the KKT, as we will presently show. For now we will just accept that the objective uses the Hessian of the Lagrangian instead of the Hessian of the objective.

We will solve the QP approximation, (8.16)-(8.17), by solving the KKT equations for this problem, which, as mentioned, are linear. These equations are given by,

$$\nabla f_a - \sum_{i=1}^{m} \lambda_i \nabla g_{i,a} = \mathbf{0} \tag{8.18}$$

$$g_{i,a} = 0 \quad for\ i = 1,\ldots,m \tag{8.19}$$

Since, for example, from (8.16),

$$\nabla f_a = \nabla f^k + \nabla_x^2 \mathrm{L}^k \Delta \mathbf{x}$$

we can also write these equations in terms of the original problem,

$$\nabla f^k + \nabla_x^2 \mathrm{L}^k \Delta \mathbf{x} - \sum_{i=1}^{m} \lambda_i \nabla g_i^k = \mathbf{0} \tag{8.20}$$

$$g_i^k + \left(\nabla g_i^k\right)^{\mathrm{T}} \Delta \mathbf{x} = 0 \quad for\ i = 1,\ldots,m \tag{8.21}$$

These are a linear set of equations we can readily solve, as shown in the example in the next section. We will want to write these equations even more concisely. If we define the following matrices and vectors,

$$\mathbf{J}^k = \begin{bmatrix} \left(\nabla g_1^k\right)^{\mathrm{T}} \\ \vdots \\ \left(\nabla g_m^k\right)^{\mathrm{T}} \end{bmatrix} \qquad \left(\mathbf{J}^k\right)^T = \begin{bmatrix} \nabla g_1^k, & \cdots, & \nabla g_m^k \end{bmatrix}$$

$$\mathbf{g}^k = \begin{bmatrix} g_1^k \\ \vdots \\ g_m^k \end{bmatrix} \qquad \nabla_x^2 \mathrm{L}^k = \nabla^2 f^k - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i^k$$

We can write (8.20)-(8.21) as,

$$\nabla_x^2 \mathrm{L}^k \Delta \mathbf{x} + \left(-\mathbf{J}^k\right)^{\mathrm{T}} \lambda = -\nabla f^k \tag{8.22}$$

$$\mathbf{J}^k \Delta \mathbf{x} = -\mathbf{g}^k \tag{8.23}$$

Again, to emphasize, this set of equations represents the solution to (8.18)-(8.19).

8.3.3 Example 1: Solving the SQP Approximation

Suppose we have as our approximation the following,

$$\begin{aligned} f_a &= 3 + \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x_1 & \Delta x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \\ g_a &= 5 + \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = 0 \end{aligned} \tag{8.24}$$

We can write out the KKT equations for this approximation as,

$$\begin{aligned} \nabla f_a - \sum_{i=1}^{m} \lambda_i \nabla g_i^k = \mathbf{0} &= \begin{bmatrix} 3 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} - \lambda \begin{bmatrix} 1 \\ 3 \end{bmatrix} = 0 \\ g_a &= 5 + \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = 0 \end{aligned} \tag{8.25}$$

We can write these equations in matrix form as,

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -3 \\ 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ -5 \end{bmatrix} \tag{8.26}$$

The solution is,

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -2.6 \\ -0.8 \\ 0.4 \end{bmatrix} \tag{8.27}$$

Observations: This calculation represents the main step in an iteration of the SQP algorithm which solves a *sequence* of *quadratic programs*. If we wanted to continue, we would add $\Delta\mathbf{x}$ to our current $\mathbf{x}$, update the Lagrangian Hessian, make a new approximation, solve for that solution, and continue iterating in this fashion.

If we ever reach a point where $\Delta\mathbf{x}$ goes to zero as we solve for the optimum of the approximation, *the original KKT equations are satisfied.* We can see this by examining (8.20)-(8.21). If $\Delta\mathbf{x}$ is zero, we have,

$$\nabla f + \underbrace{\nabla_x^2 \mathrm{L} \Delta\mathbf{x}}_{=0} - \sum_{i=1}^{m} \lambda_i^* \nabla g_i = \mathbf{0} \tag{8.28}$$

$$g_i + \underbrace{\left(\nabla g_i\right)^{\mathrm{T}} \Delta \mathbf{x}}_{=0} = 0 \quad for\ i = 1,\ldots,m \tag{8.29}$$

which then match (8.18)-(8.19).

8.3.4 NR on the KKT Equations for Problems with Equality Constraints

In this section we wish to look at applying the NR method to the original KKT equations. The KKT equations for a problem with equality constraints only are,

$$\nabla f - \sum_{i=1}^{m} \lambda_i \nabla g_i = \mathbf{0} \tag{8.30}$$

$$g_i = 0 \quad i = 1,\ldots,m \tag{8.31}$$

Now suppose we wish to solve these equations using the NR method. The coefficient matrix for NR will be composed of the derivatives of (8.30)-(8.31). For example, the first row of the coefficient matrix would be,

$$\begin{bmatrix} \left(\nabla_x f_{1NR}\right)^T & \left(\nabla_\lambda f_{1NR}\right)^T \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} f_{1NR} \\ f_{2NR} \end{bmatrix} \tag{8.32}$$

where function f_{1NR} given by,

$$f_{1NR} = \frac{\partial f}{\partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_1} \tag{8.33}$$

If we substitute (8.33) into (8.32), the first row becomes,

$$\left[\frac{\partial^2 f}{\partial x_1^2} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_1^2}\right], \left[\frac{\partial^2 f}{\partial x_2 \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_2 \partial x_1}\right], \ldots, \left[\frac{\partial^2 f}{\partial x_n \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_n \partial x_1}\right], \left[-\frac{\partial g_1}{\partial x_1}\right], \left[-\frac{\partial g_2}{\partial x_1}\right], \ldots, \left[-\frac{\partial g_m}{\partial x_1}\right]$$

Recalling,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i$$

And using the matrices,

$$\mathbf{J}^k = \begin{bmatrix} \left(\nabla g_1^k\right)^{\mathrm{T}} \\ \vdots \\ \left(\nabla g_m^k\right)^{\mathrm{T}} \end{bmatrix} \qquad \left(\mathbf{J}^k\right)^T = \begin{bmatrix} \nabla g_1^k, & \cdots, & \nabla g_m^k \end{bmatrix}$$

$$\mathbf{g}^k = \begin{bmatrix} g_1^k \\ \vdots \\ g_m^k \end{bmatrix} \qquad \nabla_x^2 \mathrm{L}^k = \nabla^2 f^k - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i^k$$

we can write the NR equations in matrix form as,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k & \left(-\mathbf{J}^k\right)^T \\ \mathbf{J}^k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} - \mathbf{x}^k \\ \lambda - \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla f^k + \left(\mathbf{J}^k\right)^T \lambda^k \\ -(\mathbf{g}^k) \end{bmatrix} \tag{8.34}$$

If we do the matrix multiplications we have

$$\begin{aligned} &\nabla_x^2 \mathrm{L}^k \Delta\mathbf{x} + \left(-\mathbf{J}^k\right)^{\mathrm{T}} (\lambda - \lambda^k) = -\nabla f^k + \left(\mathbf{J}^k\right)^T \lambda^k \\ &\mathbf{J}^k \Delta\mathbf{x} = -\left(\mathbf{g}^k\right) \end{aligned} \tag{8.35}$$

and collecting terms,

$$\begin{aligned} &\nabla_x^2 \mathrm{L}^k \Delta\mathbf{x} + \left(-\mathbf{J}^k\right)^{\mathrm{T}} \lambda = -\nabla f^k \\ &\mathbf{J}^k \Delta\mathbf{x} = -\left(\mathbf{g}^k\right) \end{aligned} \tag{8.36}$$

which equations are the same as (8.22)-(8.23)). Thus we see that *solving for the optimum of the QP approximation is the same as doing a NR iteration on the KKT equations.* This is the reason we use the Hessian of the Lagrangian function rather than the Hessian of the objective in the approximation.

8.3.5 SQP: Inequality and Equality Constraints

In the previous section we considered equality constraints only. We need to extend these results to the general case. We will state this problem as

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \geq 0 \quad i = 1,\ldots,k \\ & g_i(\mathbf{x}) = 0 \quad i = k+1,\ldots,m \end{aligned} \tag{8.37}$$

The quadratic approximation at point $\mathbf{x}^k$ is:

$$\begin{aligned} \text{Min} \quad & f_a = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \Delta\mathbf{x} + \frac{1}{2}\left(\Delta\mathbf{x}\right)^{\mathrm{T}} \nabla_x^2 \mathrm{L}^k \Delta\mathbf{x} \\ \text{s.t.} \quad & g_{i,a} : g_i^k + \left(\nabla g_i^k\right)^T \Delta\mathbf{x} \geq 0 \quad i = 1,2,\ldots,k \end{aligned} \tag{8.38}$$

$$g_i^k + \left(\nabla g_i^k\right)^T \Delta \mathbf{x} = 0 \quad i = k+1,...,m$$

Notice that the approximations are a function only of $\Delta\mathbf{x}$**.** All gradients and the Lagrangian hessian in (7.40) are evaluated at the point of expansion and so represent known quantities.

In the article where Powell [24] describes this algorithm he makes a significant statement at this point. Quoting, "The extension of the Newton iteration to take account of inequality constraints on the variables arises from the fact that the value of $\Delta\mathbf{x}$ that solves (8.36) can also be found by solving a quadratic programming problem. Specifically, $\Delta\mathbf{x}$ is the value that makes the quadratic function in (8.38) stationary."

Further, the value of $\boldsymbol{\lambda}$ for the KKT conditions is equal to the vector of Lagrange multipliers of the quadratic programming problem. Thus solving the quadratic objective and linear constraints in (8.38) is the *same as solving the NR iteration on the original KKT equations.*

The main difficulty in extending SQP to the general problem has to do the with the complementary slackness condition. This equation is nonlinear, and so makes the QP problem nonlinear. We recall that complementary slackness basically enforces that either a constraint is binding or the associated Lagrange multiplier is zero. Thus we can incorporate this condition if we can develop a method to *determine which inequality constraints are binding at the optimum*. An example of such a solution technique is given by Goldfarb and Idnani [25]. This algorithm starts out by solving for the unconstrained optimum to the problem and evaluating which constraints are violated. It then moves to add in these constraints until it is at the optimum. Thus it tends to drive to the optimum from infeasible space.

There are other important details to develop a realistic, efficient SQP algorithm. For example, the QP approximation involves the Lagrangian hessian matrix, which involves second derivatives. As you might expect, we don't evaluate the Hessian directly but approximate it using a quasi-Newton update, such as the BFGS update.

Recall that updates use differences in **x** and differences in gradients to estimate second derivatives. To estimate $\nabla_x^2 \mathrm{L}$ we will need to use differences in the gradient of the Lagrangian function,

$$\nabla_x \mathrm{L} = \nabla f - \sum_{i=1}^{m} \lambda_i \nabla g_i$$

Note that to evaluate this gradient we need values for λ_i. We will get these from our solution to the QP problem. Since our update stays positive definite, we don't have to worry about the method diverging like Newton's method does for unconstrained problems.

8.3.6 Comments on the SQP Algorithm

The SQP algorithm has the following characteristics,

- The algorithm is usually very fast.

- Because it does not rely on a traditional line search, it is often more accurate in identifying an optimum.
- The efficiency of the algorithm is partly because it does not enforce feasibility of the constraints at each step. Rather it gradually enforces feasibility as part of the KKT conditions. It is only guaranteed to be feasible at the optimum.

Relative to engineering problems, there are some drawbacks:

- Because it can go infeasible during optimization—sometimes by relatively large amounts—it can crash engineering models.
- It is more sensitive to numerical noise and/or error in derivatives than GRG.
- If we terminate the optimization process before the optimum is reached, SQP does not guarantee that we will have in-hand a better design than we started with.

8.3.7 Summary of Steps for SQP Algorithm

1. Make a QP approximation to the original problem. For the first iteration, use a Lagrangian Hessian equal to the identity matrix.

2. Solve for the optimum to the QP problem. As part of this solution, values for the Lagrange multipliers are obtained.

3. Execute a simple line search by first stepping to the optimum of the QP problem. So the initial step is $\Delta\mathbf{x}$, and $\mathbf{x}^{new} = \mathbf{x}^{old} + \Delta\mathbf{x}$. See if at this point a penalty function, composed of the values of the objective and violated constraints, is reduced. If not, cut back the step size until the penalty function is reduced. The penalty function is given by $P = f + \sum_{i=1}^{vio} \lambda_i |g_i|$ where the summation is done over the set of violated constraints, and the absolute values of the constraints are taken. The Lagrange multipliers act as scaling or weighting factors between the objective and violated constraints.

4. Evaluate the Lagrangian gradient at the new point. Calculate the difference in $\mathbf{x}$ and in the Lagrangian gradient, $\boldsymbol{\gamma}$. Update the Lagrangian Hessian using the BFGS update.

5. Return to Step 1 until $\Delta\mathbf{x}$ is sufficiently small. When $\Delta\mathbf{x}$ approaches zero, the KKT conditions for the original problem are satisfied.

8.3.8 Example of SQP Algorithm

Find the optimum to the problem,

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5 \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 \geq 0 \end{aligned}$$

starting from the point $[-1,4]$. A contour plot of the problem is shown in Fig. 8.2. This is similar to Rosenbrock's function with a constraint. The problem is interesting for several reasons: the objective is quite eccentric at the optimum, the algorithm starts at a point where

the search direction is pointing away from the optimum, and the constraint boundary at the starting point has a slope opposite to that at the optimum.

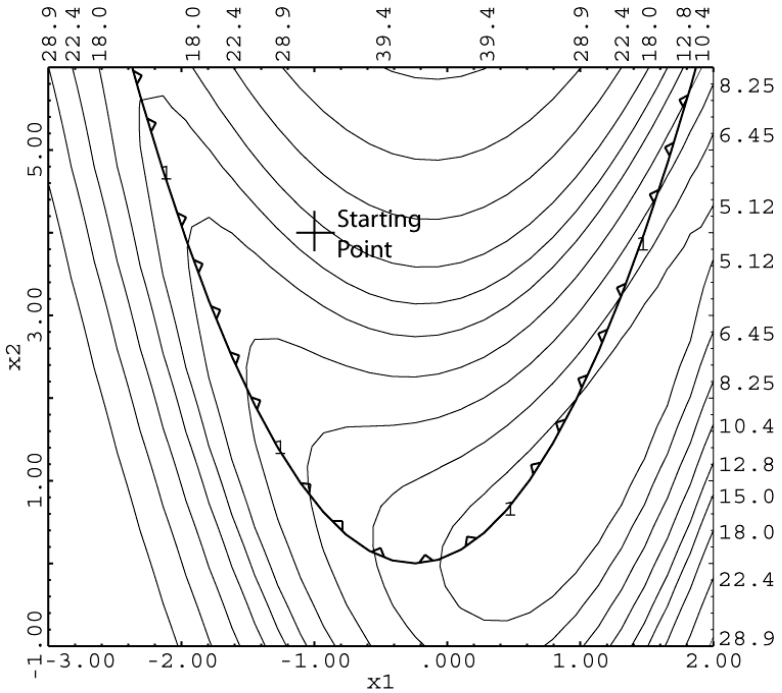

Fig. 8.2. Contour plot of example problem for SQP algorithm.

Iteration 1

We calculate the gradients, etc. at the beginning point. The Lagrangian Hessian is initialized to the identity matrix.

$$\text{At } \left(\mathbf{x}^0\right)^{\mathrm{T}} = [-1, 4],\ f^0 = 17,\ \left(\nabla f^0\right)^T = [8, 6],\ \nabla^2 \mathrm{L}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$g^0 = 2.4375,\ \left(\nabla g^0\right)^T = [1.5,\ 0.75]$$

Based on these values, we create the first approximation,

$$f_a = 17.0 + \begin{bmatrix} 8 & 6 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x_1 & \Delta x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

$$g_a = 2.4375 + \begin{bmatrix} 1.5 & 0.75 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \geq 0$$

We will assume the constraint is binding. Then the KKT conditions for the optimum of the approximation are given by the following equations:

$$\nabla f_a - \lambda \nabla g_a = 0$$

$$g_a = 0$$

These equations can be written as,

$$\begin{aligned} &8 + \Delta x_1 - \lambda(1.5) = 0 \\ &6 + \Delta x_2 - \lambda(0.75) = 0 \\ &2.4375 + 1.5\Delta x_1 + 0.75\Delta x_2 = 0 \end{aligned}$$

The solution to this set of equations is $\Delta x_1 = -0.5,\ \Delta x_2 = -2.25,\ \lambda = 5.00$

The proposed step is, $\mathbf{x}^1 = \mathbf{x}^0 + \Delta\mathbf{x} = \begin{bmatrix} -1 \\ 4 \end{bmatrix} + \begin{bmatrix} -0.5 \\ -2.25 \end{bmatrix} = \begin{bmatrix} -1.5 \\ 1.75 \end{bmatrix}$

Before we accept this step, however, we need to check the penalty function,

$$P = f + \sum_{i=1}^{vio} \lambda_i |g_i|$$

to make sure it decreased with the step. At the starting point, the constraint is satisfied, so the penalty function is just the value of the objective, $P = 17$. At the proposed point the objective value is $f = 10.5$ and the constraint is slightly violated with $g = -0.25$. The penalty function is therefore, $P = 10.5 + 5.0 * |-0.25| = 11.75$. Since this is less than 17, we accept the full step. Contours of the first approximation and the path of the first step are shown in Fig. 8.3.

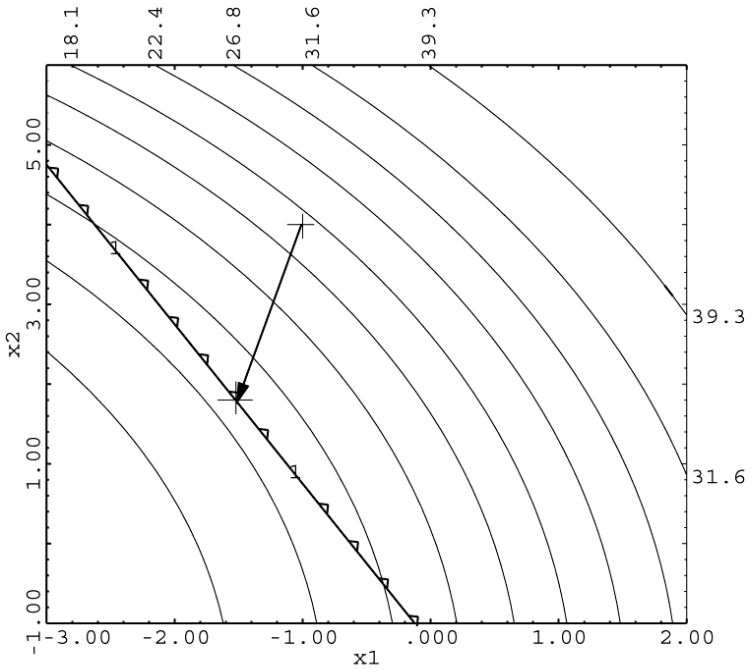

Fig. 8.3 The first SQP approximation and step.

Iteration 2

At $\left(\mathbf{x}^1\right)^{\mathrm{T}} = \begin{bmatrix} -1.5 & 1.75 \end{bmatrix}$, $f^1 = 10.5$, $\left(\nabla f^1\right)^T = \begin{bmatrix} -8.0 & -1.0 \end{bmatrix}$,

$g^1 = -0.25$, $\left(\nabla g^1\right)^T = \begin{bmatrix} 2.5 & 0.75 \end{bmatrix}$

We now need to update the Hessian of the Lagrangian. To do this we need the Lagrangian gradient at $\mathbf{x}^0$ and $\mathbf{x}^1$. (Note that we use the same Lagrange multiplier, λ^1, for both gradients.)

$$\nabla \mathrm{L}\left(\mathbf{x}^0, \boldsymbol{\lambda}^1\right) = \begin{bmatrix} 8.0 \\ 6.0 \end{bmatrix} - (5.0)\begin{bmatrix} 1.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 2.25 \end{bmatrix}$$

$$\nabla \mathrm{L}\left(\mathbf{x}^1, \boldsymbol{\lambda}^1\right) = \begin{bmatrix} -8.0 \\ -1.0 \end{bmatrix} - (5.0)\begin{bmatrix} 2.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -20.5 \\ -4.75 \end{bmatrix}$$

$$\boldsymbol{\gamma}^0 = \nabla \mathrm{L}\left(\mathbf{x}^1, \boldsymbol{\lambda}^1\right) - \nabla \mathrm{L}\left(\mathbf{x}^0, \boldsymbol{\lambda}^1\right) = \begin{bmatrix} -21.0 \\ -7.0 \end{bmatrix}$$

$$\Delta \mathbf{x}^0 = \begin{bmatrix} -1.5 \\ 1.75 \end{bmatrix} - \begin{bmatrix} -1.0 \\ 4.0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -2.25 \end{bmatrix}$$

From Chapter 3, we will use the BFGS Hessian update,

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{\boldsymbol{\gamma}^k \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}}{\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}} \Delta \mathbf{x}^k} - \frac{\mathbf{H}^k \Delta \mathbf{x}^k \left(\Delta \mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}^k}{\left(\Delta \mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}^k \Delta \mathbf{x}^k}$$

Substituting:

$$\nabla^2 \mathrm{L}^1 = \begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix} + \frac{\begin{bmatrix} -21.0 \\ -7.0 \end{bmatrix}\begin{bmatrix} -21.0 & -7.0 \end{bmatrix}}{\begin{bmatrix} -21.0 & -7.0 \end{bmatrix}\begin{bmatrix} -0.5 \\ -2.25 \end{bmatrix}} - \frac{\begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix}\begin{bmatrix} -0.5 \\ -2.25 \end{bmatrix}\begin{bmatrix} -0.5 & -2.25 \end{bmatrix}\begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix}}{\begin{bmatrix} -0.5 & -2.25 \end{bmatrix}\begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix}\begin{bmatrix} -0.5 \\ -2.25 \end{bmatrix}}$$

$$\nabla^2 \mathrm{L}^1 = \begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix} + \begin{bmatrix} 16.8000 & 5.6000 \\ 5.6000 & 1.8667 \end{bmatrix} - \begin{bmatrix} 0.0471 & 0.2118 \\ 0.2118 & 0.9529 \end{bmatrix}$$

$$\nabla^2 \mathrm{L}^1 = \begin{bmatrix} 17.7529 & 5.3882 \\ 5.3882 & 1.9137 \end{bmatrix}$$

The second approximation is therefore,

$$f_a = 10.5 + \begin{bmatrix} -8.0 & -1.0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x_1 & \Delta x_2 \end{bmatrix} \begin{bmatrix} 17.753 & 5.3882 \\ 5.3882 & 1.9137 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

$$g_a = -0.25 + \begin{bmatrix} 2.5 & 0.75 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \geq 0$$

As we did before, we will assume the constraint is binding. The KKT equations are,

$$\begin{aligned} &-8 + 17.753\Delta x_1 + 5.3882\Delta x_2 - \lambda(2.5) = 0 \\ &-1 + 5.3882\Delta x_1 + 1.9137\Delta x_2 - \lambda(0.75) = 0 \\ &-0.25 + 2.5\Delta x_1 + 0.75\Delta x_2 = 0 \end{aligned}$$

The solution to this set of equations is $\Delta x_1 = 1.6145,\ \Delta x_2 = -5.048,\ \lambda = -2.615$. Because λ is negative, we need to drop the constraint from the picture. (We can see in Fig. 8.4 below that the constraint is not binding at the optimum.) With the constraint dropped, the solution is, $\Delta x_1 = 2.007,\ \Delta x_2 = -5.131,\ \lambda = 0$. This gives a new **x** of,

$$\mathbf{x}^2 = \mathbf{x}^1 + \Delta\mathbf{x} = \begin{bmatrix} -1.5 \\ 1.75 \end{bmatrix} + \begin{bmatrix} 2.007 \\ -5.131 \end{bmatrix} = \begin{bmatrix} 0.507 \\ -3.381 \end{bmatrix}$$

However, when we try to step this far, we find the penalty function has increased from 11.75 to 17.48 (this is the value of the objective only—the violated constraint does not enter in to the penalty function because the Lagrange multiplier is zero). We cut the step back. How much to cut back is somewhat arbitrary. We will make the step 0.5 times the original. The new value of **x** becomes,

$$\mathbf{x}^2 = \mathbf{x}^1 + \Delta\mathbf{x} = \begin{bmatrix} -1.5 \\ 1.75 \end{bmatrix} + 0.5 \begin{bmatrix} 2.007 \\ -5.131 \end{bmatrix} = \begin{bmatrix} -0.4965 \\ -0.8155 \end{bmatrix}$$

At which point the penalty function is 7.37. So we accept this step. Contours of the second approximation are shown in Fig. 8.4, along with the step taken.

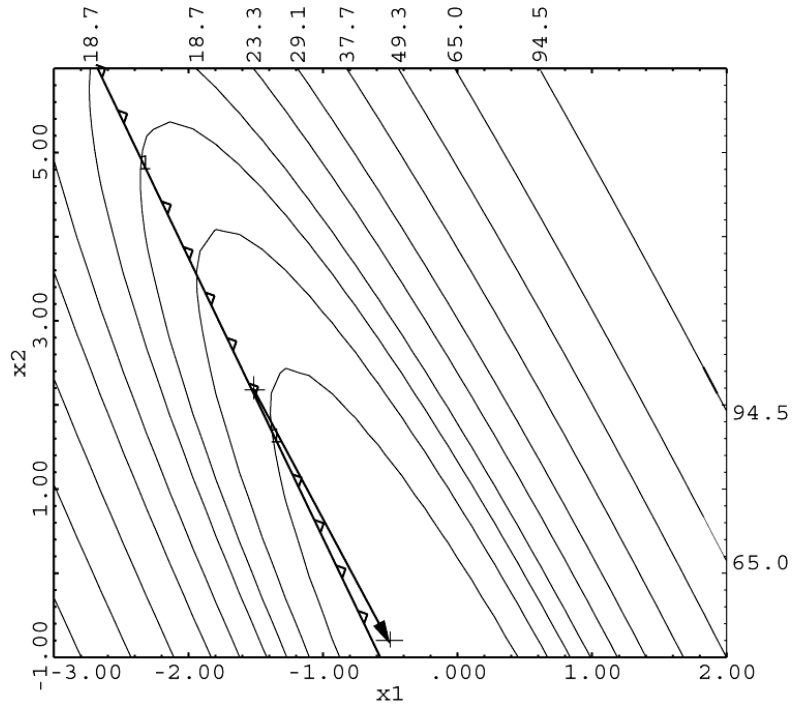

Fig. 8.4 The second approximation and step.

Iteration 3

At $\left(\mathbf{x}^2\right)^{\mathrm{T}} = \begin{bmatrix} -0.4965 & -0.8155 \end{bmatrix}$, $f^2 = 7.367$, $\left(\nabla f^2\right)^T = \begin{bmatrix} -5.102 & -2.124 \end{bmatrix}$,

$g^2 = -0.6724$, $\left(\nabla g^2\right)^T = \begin{bmatrix} 0.493 & 0.75 \end{bmatrix}$

$$\nabla \mathrm{L}\left(\mathbf{x}^1, \boldsymbol{\lambda}^2\right) = \begin{bmatrix} -8.0 \\ -1.0 \end{bmatrix} - (0) \begin{bmatrix} 2.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -8.0 \\ -1.0 \end{bmatrix}$$

$$\nabla \mathrm{L}\left(\mathbf{x}^2, \boldsymbol{\lambda}^2\right) = \begin{bmatrix} -5.102 \\ -2.124 \end{bmatrix} - (0) \begin{bmatrix} 0.493 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -5.102 \\ -2.124 \end{bmatrix}$$

$$\boldsymbol{\gamma}^1 = \nabla \mathrm{L}\left(\mathbf{x}^2, \boldsymbol{\lambda}^2\right) - \nabla \mathrm{L}\left(\mathbf{x}^1, \boldsymbol{\lambda}^2\right) = \begin{bmatrix} 2.898 \\ -1.124 \end{bmatrix}$$

$$\Delta \mathbf{x}^1 = \begin{bmatrix} -0.4965 \\ -0.8155 \end{bmatrix} - \begin{bmatrix} -1.5 \\ 1.75 \end{bmatrix} = \begin{bmatrix} 1.004 \\ -2.5655 \end{bmatrix}$$

Based on these vectors, the new Lagrangian Hessian is,

$$\nabla^2 \mathrm{L}^2 = \begin{bmatrix} 17.7529 & 5.3882 \\ 5.3882 & 1.9137 \end{bmatrix} + \begin{bmatrix} 1.4497 & -0.5623 \\ -0.5623 & 0.2181 \end{bmatrix} - \begin{bmatrix} 5.8551 & 0.7320 \\ 0.7320 & 0.0915 \end{bmatrix}$$

$$\nabla^2 L^2 = \begin{bmatrix} 13.3475 & 4.0939 \\ 4.0939 & 2.0403 \end{bmatrix}$$

So our next approximation is,

$$f_a = 7.367 + \begin{bmatrix} -5.102 & -2.124 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x_1 & \Delta x_2 \end{bmatrix} \begin{bmatrix} 13.3475 & 4.0939 \\ 4.0939 & 2.0403 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

$$g_a = -0.6724 + \begin{bmatrix} 0.493 & 0.75 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \geq 0$$

The KKT equations, assuming the constraint is binding, are,

$$\begin{aligned} -5.102 + 13.3475\Delta x_1 + 4.0939\Delta x_2 - \lambda(0.493) &= 0 \\ -2.124 + 4.0939\Delta x_1 + 2.0403\Delta x_2 - \lambda(0.75) &= 0 \\ -0.6724 + 0.493\Delta x_1 + 0.75\Delta x_2 &= 0 \end{aligned}$$

The solution to this set of equations is $\Delta x_1 = 0.1399,\ \Delta x_2 = 0.8046,\ \lambda = 0.1205$.

Our new proposed point is, $\mathbf{x}^3 = \mathbf{x}^2 + \Delta\mathbf{x} = \begin{bmatrix} -0.4965 \\ -0.8155 \end{bmatrix} + \begin{bmatrix} 0.1399 \\ 0.8046 \end{bmatrix} = \begin{bmatrix} -0.3566 \\ -0.0109 \end{bmatrix}$

At this point the penalty function has decreased from 7.37 to 5.85. We accept the full step. A contour plot of the third approximation is shown in Fig. 8.5.

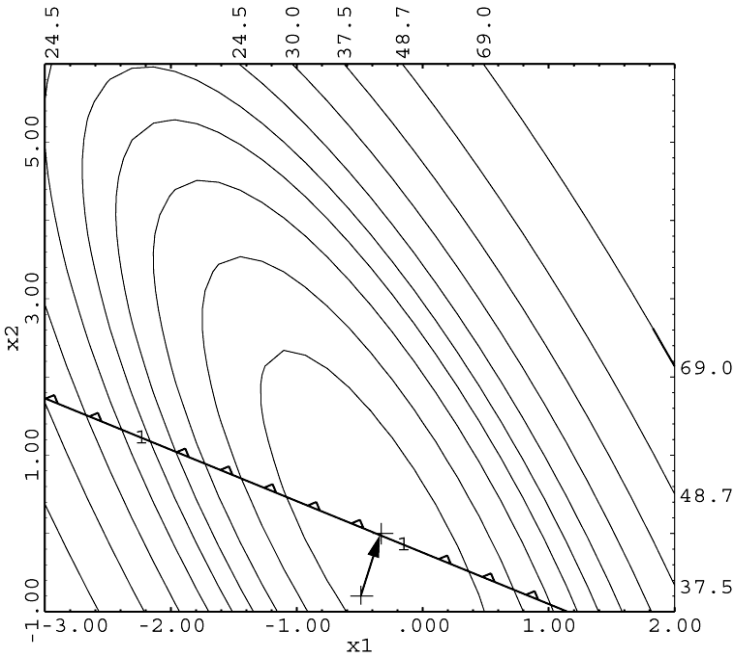

Fig. 8.5 The third approximation and step.

Iteration 4

At $\left(\mathbf{x}^3\right)^T = \begin{bmatrix} -0.3566 & -0.0109 \end{bmatrix}$, $f^3 = 5.859$, $\left(\nabla f^3\right)^T = \begin{bmatrix} -2.9101 & -0.2761 \end{bmatrix}$,

$g^3 = -0.01954$, $\left(\nabla g^3\right)^T = \begin{bmatrix} 0.2132 & 0.75 \end{bmatrix}$

$$\nabla \mathrm{L}\left(\mathbf{x}^2, \boldsymbol{\lambda}^3\right) = \begin{bmatrix} -5.102 \\ -2.124 \end{bmatrix} - (0.1205)\begin{bmatrix} 0.493 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -5.161 \\ -2.214 \end{bmatrix}$$

$$\nabla \mathrm{L}\left(\mathbf{x}^3, \boldsymbol{\lambda}^3\right) = \begin{bmatrix} -2.910 \\ -0.2761 \end{bmatrix} - (0.1205)\begin{bmatrix} 0.2132 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -2.936 \\ -0.3665 \end{bmatrix}$$

$$\boldsymbol{\gamma}^2 = \nabla \mathrm{L}\left(\mathbf{x}^3, \boldsymbol{\lambda}^3\right) - \nabla \mathrm{L}\left(\mathbf{x}^2, \boldsymbol{\lambda}^3\right) = \begin{bmatrix} 2.225 \\ 1.8475 \end{bmatrix}$$

$$\Delta \mathbf{x}^2 = \begin{bmatrix} -0.3566 \\ -0.0109 \end{bmatrix} - \begin{bmatrix} -0.4965 \\ -0.8155 \end{bmatrix} = \begin{bmatrix} 0.1399 \\ 0.8046 \end{bmatrix}$$

Based on these vectors, the new Lagrangian Hessian is,

$$\nabla^2 \mathrm{L}^3 = \begin{bmatrix} 13.3475 & 4.0939 \\ 4.0939 & 2.0403 \end{bmatrix} + \begin{bmatrix} 2.7537 & 2.2865 \\ 2.2865 & 1.8986 \end{bmatrix} - \begin{bmatrix} 10.6397 & 4.5647 \\ 4.5647 & 1.9584 \end{bmatrix}$$

$$\nabla^2 \mathrm{L}^3 = \begin{bmatrix} 5.4616 & 1.8157 \\ 1.8157 & 1.9805 \end{bmatrix}$$

Our new approximation is,

$$f_a = 5.859 + \begin{bmatrix} -2.910 & -0.2761 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \Delta x_1 & \Delta x_2 \end{bmatrix}\begin{bmatrix} 5.4616 & 1.8157 \\ 1.8157 & 1.9805 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

$$g_a = -0.0195 + \begin{bmatrix} 0.2132 & 0.75 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \geq 0$$

The KKT equations, assuming the constraint is binding, are,

$$\begin{aligned} &-2.910 + 5.4616\Delta x_1 + 1.8157\Delta x_2 - \lambda(0.2132) = 0 \\ &-0.2761 + 1.8157\Delta x_1 + 1.9805\Delta x_2 - \lambda\left(0.75\right) = 0 \\ &-0.0195 + 0.2132\Delta x_1 + 0.75\Delta x_2 = 0 \end{aligned}$$

The solution to this problem is, $\Delta x_1 = 0.6099,\ \Delta x_2 = -0.1474,\ \lambda = 0.7192$. Since λ is positive, our assumption about the constraint was correct. Our new proposed point is,

$$\mathbf{x}^4 = \mathbf{x}^3 + \Delta\mathbf{x} = \begin{bmatrix} -0.3566 \\ -0.0109 \end{bmatrix} + \begin{bmatrix} 0.6099 \\ -0.1474 \end{bmatrix} = \begin{bmatrix} 0.2533 \\ -0.1583 \end{bmatrix}$$

At this point the penalty function is 4.87, a decrease from 5.85, so we take the full step. The contour plot is given in Fig. 8.6. The fifth step is shown in Fig. 8.7.

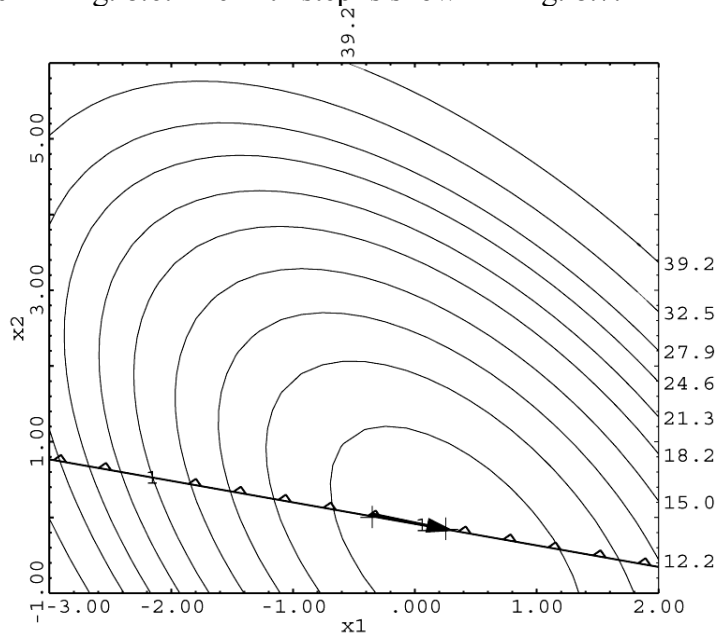

Fig. 8.6 The fourth approximation and step.

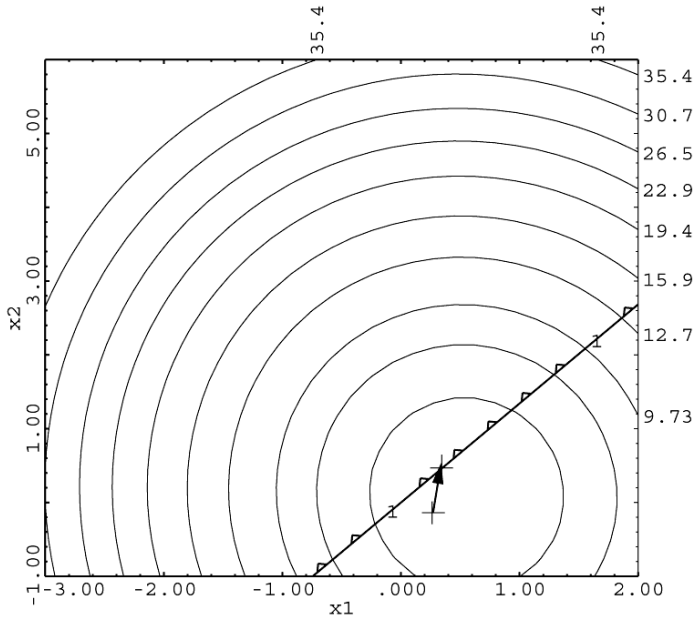

Fig. 8.7 The fifth approximation and step.

We would continue in this fashion until $\Delta\mathbf{x}$ goes to zero. We would then know the original KKT equations were satisfied. The solution to this problem occurs at,

$$\left(\mathbf{x}^*\right)^{\mathrm{T}} = \begin{bmatrix} 0.500 & 0.750 \end{bmatrix}, \quad f^* = 4.50$$

A summary of five steps is overlaid on the original problem in Fig. 8.8.

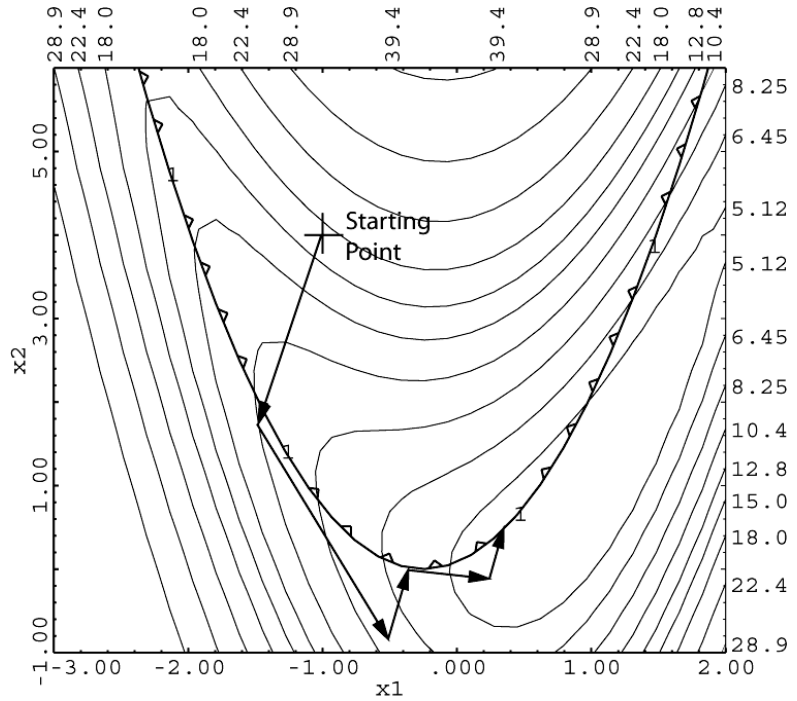

Fig. 8.8 The path of the SQP algorithm.

8.4 The Interior Point (IP) Algorithm

8.4.1 Problem Definition

In general, for the *primal-dual* method given here, there are two approaches to developing the equations for the IP method: putting slack variables into their own vector, and incorporating slack variables as part of the design variables **x**. Both methods can be found in the literature. We have opted for the former, because the development is somewhat more straightforward. It is also the approach used in MATLAB. The development of the latter is given in a later optional section.

For simplicity, we will start with a problem that only has inequality constraints.

$$\text{Min} \quad f(\mathbf{x}) \tag{8.39}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) \geq 0 \quad i = 1,\ldots,m \tag{8.40}$$

We will add in *slack* variables to turn the inequalities into equalities:

$$\text{Min} \quad f(\mathbf{x}) \tag{8.41}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) - s_i = 0 \quad i = 1,\ldots,m \tag{8.42}$$

$$s_i \geq 0 \quad i = 1,...,m \tag{8.43}$$

Slack variables are so named because they take up the "slack" between the constraint value and the right hand side. For an inequality constraint to be feasible, s_i must be ≥ 0. If (8.42) is satisfied and $s_i = 0$, then the original inequality is binding.

The IP algorithm eliminates the lower bounds on **s** by incorporating a *barrier function* as part of the objective:

$$\text{Min} \quad f_\mu = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(s_i) \tag{8.44}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) - s_i = 0 \quad i = 1,\ldots,m \tag{8.45}$$

We note that as s_i approaches zero (from a positive value, i.e. from feasible space), the negative barrier term goes to infinity. This obviously penalizes the objective and forces the algorithm to keep *s* positive. The IP algorithm solves a sequence of these problems for a decreasing set of barrier parameters μ. As μ approaches zero, the barrier becomes steeper and sharper. This is illustrated in Fig. 8.9.

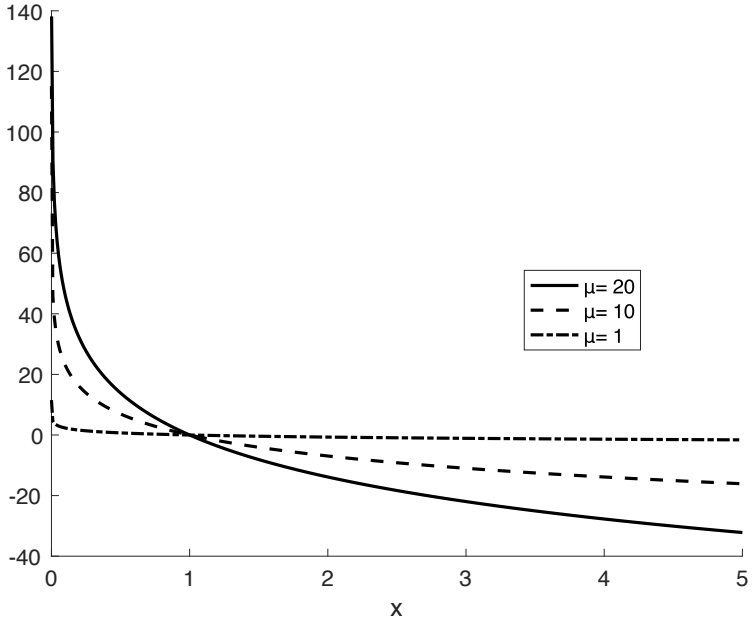

Fig. 8.9 Barrier function $-\mu \ln(x)$ for μ=20, 10 and 1.

We can define the Lagrangian function for this problem as,

$$L(\mathbf{x},\mathbf{s},\lambda) = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(s_i) - \sum_{i=1}^{m} \lambda_i (g_i(\mathbf{x}) - s_i) \tag{8.46}$$

Taking the gradient of this function with respect to $\mathbf{x}, \mathbf{s}, \lambda$, and setting it equal to zero gives us the KKT conditions for (8.44)–(8.45),

$$\nabla_x L = \nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) = \mathbf{0} \tag{8.47}$$

$$\nabla_s L = s_i \lambda_i - \mu = 0 \qquad i = 1,\ldots,m \tag{8.48}$$

$$\nabla_\lambda L = -(g_i(\mathbf{x}) - s_i) = 0 \qquad i = 1,\ldots,m \tag{8.49}$$

If we define **e** as the vector of 1's of *m* dimension and,

$$\mathbf{S} = \begin{pmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_m \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix}$$

we can replace (8.48) above with (8.51) below,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) = \mathbf{0} \tag{8.50}$$

$$\mathbf{S\Lambda e} - \mu \mathbf{e} = \mathbf{0} \tag{8.51}$$

$$g_i(\mathbf{x}) - s_i = 0 \quad i = 1,\ldots,m \tag{8.52}$$

It is worth noting that as $\mu \to 0$, the above equations (along with $\lambda, \mathbf{s} \geq 0$) represent the KKT conditions for the original problem (8.39)-(8.40). Notice that with $\mu = 0$, (8.51) (perhaps more easily seen in (8.48)) expresses complementary slackness for the slack variable bounds, i.e. either $\lambda_i = 0$ or $s_i = 0$.

8.4.2 Problem Solution

We will use the NR method to solve the set of equations represented by (8.50)-(8.52). The coefficient matrix (see 8.5) will be the first derivatives of these equations. We note that we have *n* equations from (8.50), *m* equations from (8.51) and *m* equations from (8.52). Similarly, these equations are functions of *n* variables *x*, *m* variables λ, and *m* variables *s*.

The coefficient matrix for NR can be represented as,

$$\begin{bmatrix} \left(\nabla_x f_{1NR}\right)^T & \left(\nabla_s f_{1NR}\right)^T & \left(\nabla_\lambda f_{1NR}\right)^T \\ \vdots & \vdots & \vdots \\ \left(\nabla_x f_{(n+2m)NR}\right)^T & \left(\nabla_s f_{(n+2m)NR}\right)^T & \left(\nabla_\lambda f_{(n+2m)NR}\right)^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} \tag{8.53}$$

where f_{1NR} (the first NR equation) is given by,

$$f_{1NR} = \frac{\partial f}{\partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_1} \tag{8.54}$$

and *r1*, *r2* and *r3* represent the residuals for (8.50)–(8.52). If we substitute (8.54) into the first row of (8.53), the first row of the coefficient matrix becomes,

$$\underbrace{\left[\frac{\partial^2 f}{\partial x_1^2} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_1^2}\right], \ldots, \left[\frac{\partial^2 f}{\partial x_n \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_n \partial x_1}\right]}_{\nabla_x f^T}, \underbrace{[0], \ldots, [0]}_{\nabla_s f^T}, \underbrace{\left[-\frac{\partial g_1}{\partial x_1}\right], \ldots, \left[-\frac{\partial g_m}{\partial x_1}\right]}_{\nabla_\lambda f^T}$$

Recalling,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i \tag{8.55}$$

and

$$\mathbf{J}^k = \begin{bmatrix} \left(\nabla g_1^k\right)^{\mathrm{T}} \\ \vdots \\ \left(\nabla g_m^k\right)^{\mathrm{T}} \end{bmatrix} \qquad \left(\mathbf{J}^k\right)^T = \begin{bmatrix} \nabla g_1^k, & \cdots & \nabla g_m^k \end{bmatrix}$$

we can write the NR iteration equations at step *k* as,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k & 0 & \left(-\mathbf{J}^k\right)^T \\ 0 & \Lambda^k & \mathbf{S}^k \\ \mathbf{J}^k & -\mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\mathbf{s}^k \\ \Delta\lambda^k \end{bmatrix} = - \begin{bmatrix} \nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) \\ \mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} \\ \mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k \end{bmatrix} \tag{8.56}$$

At this point, we could solve this system of equations to obtain $\Delta\mathbf{x}^k$, $\Delta\mathbf{s}^k$, and $\Delta\boldsymbol{\lambda}^k$. However, for large problems we can gain efficiency by simplifying this expression. If we look at row 2 above,

$$\Lambda^k \Delta\mathbf{s}^k + \mathbf{S}^k \Delta\lambda^k = -\mathbf{S}^k \Lambda^k \mathbf{e} + \mu\mathbf{e} \tag{8.57}$$

We would like to solve (8.57) for $\Delta\mathbf{s}^k$. Rearranging terms and pre-multiplying both sides by $\left(\Lambda^k\right)^{-1}$ gives,

$$\Delta\mathbf{s}^k = -\left(\Lambda^k\right)^{-1} \mathbf{S}^k \Lambda\mathbf{e} + \mu\Lambda^{-1}\mathbf{e} - \Lambda^{-1}\mathbf{S}^k \Delta\mathbf{x}^k \tag{8.58}$$

For diagonal matrices, the order of matrix multiplication does not matter, so the first term on the right hand side simplifies to give,

$$\Delta\mathbf{s}^k = -\mathbf{S}^k\mathbf{e} + \mu\Lambda^{-1}\mathbf{e} - \Lambda^{-1}\mathbf{S}^k \Delta\mathbf{x}^k \tag{8.59}$$

Examining the first and second terms on the right hand side, we note that,

$$-\mathbf{S}^k\mathbf{e} = -\begin{pmatrix} s_1^k & & 0 \\ & \ddots & \\ 0 & & s_n^k \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -\mathbf{s}^k \qquad \mu(\Lambda^k)^{-1}\mathbf{e} = \begin{pmatrix} \frac{\mu}{\lambda_1^k} & & 0 \\ & \ddots & \\ 0 & & \frac{\mu}{\lambda_m^k} \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{s}^k$$

Equation (8.59) becomes,

$$\begin{aligned}\Delta\mathbf{s}^k &= -\mathbf{s}^k + \mathbf{s}^k - (\Lambda^k)^{-1}\mathbf{S}^k\Delta\mathbf{x}^k \\ &= -(\Lambda^k)^{-1}\mathbf{S}^k\Delta\mathbf{x}^k\end{aligned} \tag{8.60}$$

We define Ω to be,

$$\Omega^k = (\Lambda^k)^{-1}\mathbf{S}^k \tag{8.61}$$

so that we have,

$$\Delta\mathbf{s}^k = -\Omega^k\Delta\lambda^k \tag{8.62}$$

We can then write (8.56) as a reduced set of equations:

$$\begin{bmatrix} -\nabla_x^2 \mathrm{L}^k & \left(\mathbf{J}^k\right)^T \\ \mathbf{J}^k & \Omega^k \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\lambda^k \end{bmatrix} = -\begin{bmatrix} -\nabla f(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i\nabla g_i(\mathbf{x}) \\ \mathbf{g}_i(\mathbf{x}) - \mathbf{s}_i \end{bmatrix} \tag{8.63}$$

This matrix represents a linear symmetric system of equations (symmetric because the off-diagonal elements are the transpose of one another), of lower dimension than (8.56) and is more efficiently solved. Once we have solved for $\Delta\lambda^k$, we can use (8.62) to get $\Delta\mathbf{s}^k$.

8.4.3 The Line Search

It is sometimes said, "the devil is in the details." That is certainly true for the line search for the IP method. Modern algorithms employ a number of line search techniques, including merit functions, trust regions and filter methods [26, 27, 28]. They include techniques to handle non-positive-definite hessians or otherwise poorly conditioned problems, or to regain feasibility. Sometimes a different step length is used for the primal variables (**x, s**) and the dual variables (λ). We will adopt that strategy here as well.

We can consider $\Delta\mathbf{x}, \Delta\mathbf{s}, \Delta\lambda,$ as the search directions for **x, s**, and $\boldsymbol{\lambda}$. We then need to determine the step size (between 0-1) in these directions,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k\Delta\mathbf{x}^k \tag{8.64}$$
$$\lambda^{k+1} = \lambda^k + \alpha^k\Delta\lambda^k \tag{8.65}$$
$$\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha^k\Delta\mathbf{s}^k \tag{8.66}$$

Similar to SQP, a straightforward method is to accept α if it results in a decrease in a merit function that combines the objective with a sum of the violated constraints, i.e.,

$$P = f^k + \nu\sum_{i=1}^{viol}|g_i|$$

where ν is a constant. We will also reduce the step length if necessary to keep a slack variable or a Lagrange multiplier positive.

8.4.4 Example 1: Two Variables, One Constraint

We will apply the IP method to the same example given for SQP in Section 8.3.8, namely,

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5 \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 \geq 0 \end{aligned}$$

We reformulate the problem using a slack variable, s_1, for the constraint:

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5 \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 - s_1 = 0 \\ & s_1 \geq 0 \end{aligned}$$

We eliminate the lower bound for s_1 by adding a barrier term,

$$\begin{aligned} \text{Min} \quad & f_\mu(\mathbf{x}) = f(\mathbf{x}) - \mu \ln(s_1) \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 - s_1 = 0 \end{aligned}$$

At the starting point we have,

$$\left(\mathbf{x}^0\right)^{\mathrm{T}} = \left[-1,4\right],\ f^0 = 17,\ \left(\nabla f^0\right)^T = \left[8,6\right],\ \ g^0 = 2.4375,\ \left(\nabla g^0\right)^T = \left[1.5,\ 0.75\right]$$

We will assume we do not have second derivatives available, so we will begin with the Hessian set to $\nabla^2 \mathrm{L}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. $\mathbf{J}^0 = [1.5,\ 0.75]$. We will also set $\mu^0 = 5$, $s_1^0 = 2.4375$, $\lambda_1^0 = 2$. This value of λ was picked to approximately satisfy (8.51), i.e. $\lambda^0 s^0 - \mu^0 = 0$.

If the merit function increases for a proposed step, we will cut the step in half and continue doing so several times.

8.4.4.1 First Iteration

Based on the data above, the coefficient matrix (refer back to (8.56)) for the first step is,

$$\begin{bmatrix} 1 & 0 & 0 & -1.5 \\ 0 & 1 & 0 & -0.75 \\ 0 & 0 & 2 & 2.4375 \\ 1.5 & 0.75 & -1 & 0 \end{bmatrix}$$

For the residual vector, we have,

$$\nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) = \begin{bmatrix} 8 \\ 6 \end{bmatrix} - 2\begin{bmatrix} 1.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} 5 \\ 4.5 \end{bmatrix}$$

$$\mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} = \begin{bmatrix} 2.4375 \end{bmatrix}\begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} 5 \end{bmatrix} = \begin{bmatrix} -0.125 \end{bmatrix}$$

$$\mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k = \begin{bmatrix} 2.4375 \end{bmatrix} - \begin{bmatrix} 2.4375 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

Thus the set of equations for our first step becomes,

$$\begin{bmatrix} 1 & 0 & 0 & -1.5 \\ 0 & 1 & 0 & -0.75 \\ 0 & 0 & 2 & 2.4375 \\ 1.5 & 0.75 & -1 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^0 \\ \Delta s^0 \\ \Delta \lambda^0 \end{bmatrix} = -\begin{bmatrix} 5 \\ 4.5 \\ -0.125 \\ 0 \end{bmatrix}$$

The solution is $\Delta \mathbf{x}^T = [-0.930,\ -2.465], \quad \Delta s = -3.244, \quad \Delta\lambda = 2.713$
Because the full step would make the slack negative, we set the step length for **x** and **s** to be 0.748 to keep the slack slightly positive. We accept the full step for λ. At this trial point the objective has decreased to 11.78 but the constraint is violated at –0.471. However the merit function has decreased from 17 to 14.00 and we accept the step. Our new point is $\mathbf{x}^T = [-1.695, 2.157], \quad s = 0.012, \quad \lambda = 4.713.$

8.4.4.2 Second Iteration

At $\left(\mathbf{x}^1\right)^{\mathrm{T}} = [-1.695,\ 2.157], \quad f^1 = 11.78, \quad \left(\nabla f^1\right)^T = \ [-10.256 \quad -1.435]$

$$g^1 = -0.4714, \quad \left(\nabla g^1\right)^T = \ [2.891 \quad 0.75]$$

We start this step by updating the Hessian of the Lagrangian. To do so, we evaluate the Lagrangian gradient at $\mathbf{x}^0$ and $\mathbf{x}^1$. We then calculate the γ and Δ**x** vectors. (As we did for SQP, we use the same Lagrange multiplier, λ^1, for both gradients.)

$$\nabla \mathrm{L}\left(\mathbf{x}^0, \lambda^1\right) = \begin{bmatrix} 8.0 \\ 6.0 \end{bmatrix} - \left(4.713\right)\begin{bmatrix} 1.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} 0.930 \\ 2.465 \end{bmatrix}$$

$$\nabla \mathrm{L}\left(\mathbf{x}^1, \lambda^1\right) = \begin{bmatrix} -10.256 \\ -1.435 \end{bmatrix} - \left(4.713\right) \begin{bmatrix} 2.891 \\ 0.75 \end{bmatrix} = \begin{bmatrix} -23.881 \\ -4.967 \end{bmatrix}$$

$$\gamma^0 = \nabla \mathrm{L}\left(\mathbf{x}^1, \lambda^1\right) - \nabla \mathrm{L}\left(\mathbf{x}^0, \lambda^1\right) = \begin{bmatrix} -24.811 \\ -7.435 \end{bmatrix}$$

$$\Delta \mathbf{x}^0 = \begin{bmatrix} -1.695 \\ 2.157 \end{bmatrix} - \begin{bmatrix} -1.0 \\ 4.0 \end{bmatrix} = \begin{bmatrix} -0.695 \\ -1.843 \end{bmatrix}$$

We use this data to obtain the new estimate of the Lagrangian Hessian using BFGS Hessian update, as we did for SQP (see the SQP example for details). The new Hessian is,

$$\nabla^2 \mathrm{L}^1 = \begin{bmatrix} 20.762 & 5.629 \\ 5.629 & 1.910 \end{bmatrix}$$

After evaluating the residuals, our next set of equations becomes,

$$\begin{bmatrix} 20.762 & 5.629 & 0 & -2.891 \\ 5.629 & 1.910 & 0 & -0.75 \\ 0 & 0 & 4.713 & 0.012 \\ 2.891 & 0.75 & -1 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^1 \\ \Delta s^1 \\ \Delta \lambda^1 \end{bmatrix} = - \begin{bmatrix} -23.881 \\ -4.970 \\ -0.943 \\ -0.484 \end{bmatrix}$$

The solution gives: $\Delta \mathbf{x}^T = [1.104,\ -3.319]$, $\Delta s = 0.218$, $\Delta \lambda = -6.797$. For λ we only take 0.693 of the step to keep $\lambda \geq 0$. With the full step for **x** and **s**, our merit function decreases from 14.0 to 10.8. Our new point is,

$$\left(\mathbf{x}^2\right)^T = [-0.592,\ -1.162], \quad s = 0.230, \quad \lambda = 0, \quad f = 8.820, \quad g = -0.988$$

Subsequent steps proceed in a similar manner. The progress of the algorithm to the optimum, with our relatively unsophisticated line search, is shown in Fig. 8.10 below. The algorithm reaches the optimum in about nine steps. At every iteration we reduce μ according to the equation $\mu^{k+1} = \dfrac{\mu^k}{5}$. For comparison purposes, the progress of the Interior Point method in `fmincon` on this problem is shown in Fig. 8.11.

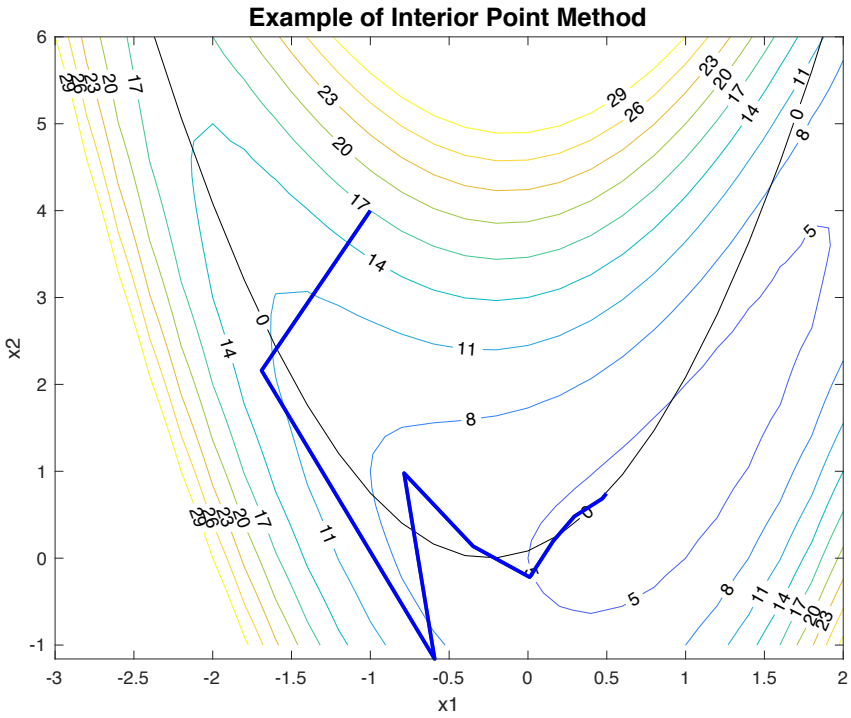

Fig 8.10. The progress of the IP algorithm on Example 1.

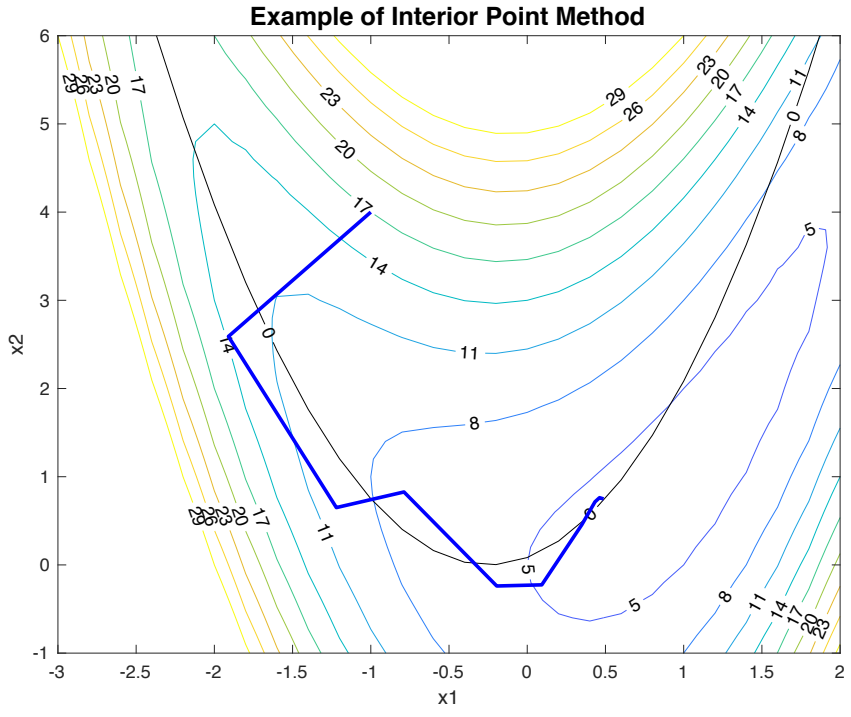

Fig. 8.11 Path of `fmincon` IP algorithm on Example 1.

8.4.5 Example 2: One Variable, Two Constraints

In this example problem the functions are very simple, but the setup for the constraints is more involved than the previous example. We will show how the problem changes when we have two slack variables.

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 \geq 0 \\ & x_1 \geq 1 \end{aligned}$$

We will change the inequality and the bound to equality constraints by means of slack variables so that we have,

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 - s_1 = 0 \\ & x_1 - 1 - s_2 = 0 \\ & s_1, s_2 \geq 0 \end{aligned}$$

We remove the bounds on the slacks by adding barrier terms,

$$\begin{aligned} \text{Min} \quad & f_\mu = x_1^2 - \mu \sum_{i=1}^{2} \ln(s_i) \\ \text{s.t.} \quad & -2x_1 + 9 - s_1 = 0 \\ & x_1 - 1 - s_2 = 0 \end{aligned}$$

By inspection, the solution to the problem is $x_1 = 1, \quad s_1 = 7, \quad s_2 = 0; \quad f = 1.$

8.4.5.1 First Iteration

At our starting point $x_1 = 3, \quad \left(\mathbf{s}^0\right)^T = [3,2], \quad f^0 = 9, \quad g_1^0 = 0, \quad g_2^0 = 0$

We also have, $\nabla f^0 = [6], \quad \nabla^2 f = [2], \quad \nabla g_1 = [-2], \quad \nabla g_2 = [1], \quad \nabla^2 g_1 = [0], \quad \nabla^2 g_2 = [0]$

We will use $\mu^0 = 2, \quad \alpha^0 = 0.5, \quad (\lambda^0)^T = [1,1]$. Thus,

$$\mathbf{S}^0 = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \quad \Lambda^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{J}^0 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Because we only have one variable, the Hessian of the Lagrangian is just a 1x1 matrix (we use the actual second derivative here for simplicity):

$$\nabla_x^2 L = \nabla^2 f - \lambda_1 \nabla^2 g_1 - \lambda_2 \nabla^2 g_2 = [2] - (1)[0] - (1)[0] = [2]$$

With this information we can then build our coefficient matrix,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k & 0 & \left(-\mathbf{J}^k\right)^T \\ 0 & \Lambda^k & \mathbf{S}^k \\ \mathbf{J}^k & -\mathbf{I} & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ -2 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

The vector of residuals is,

$$\nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) = \quad [6] - (1)[-2] - (1)[1] = 7$$

$$\mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2.0 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We can then solve for our new point as (recall alpha = 0.5),

$x_1^1 = 2.174, \quad \left(\mathbf{s}^1\right)^T = [4.652, 1.174], \quad \left(\lambda^1\right)^T = [0.283, 1.413]$ at which point $f = 4.73$, and

$g_1^1 = 0, \quad g_2^1 = 0$.

8.4.6 *An Alternate Development of the Newton Iteration Equations

*This section is optional.

As mentioned at the start of the section on IP algorithms, there are two approaches to developing the NR equations: separating out the slack variables in their own vector, and including the slacks as part of the **x** vector. Previously we kept the slacks separate. Now we will combine them with **x.** This development follows the work by Wachter and Biegler [29, 14].

For simplicity, we will start with a problem which only has equality constraints. We will, however, include a lower bound on the variables:

$$\text{Min} \quad f(\mathbf{x}) \tag{8.67}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) = 0 \quad i = 1, \ldots, m \tag{8.68}$$

$$\mathbf{x} \geq 0 \tag{8.69}$$

As before, we eliminate the lower bounds by including them in the objective function with a barrier function:

$$\text{Min} \quad f_\mu = f(\mathbf{x}) - \mu \sum_{i=1}^{n} \ln(x_i) \tag{8.70}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) = 0 \quad i = 1,\ldots,m \tag{8.71}$$

We now consider the necessary conditions for a solution to the barrier problem represented by (8.70)-(8.71). We first define,

$$z_i = \frac{\mu}{x_i} \tag{8.72}$$

We can write the KKT conditions as,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) - \mathbf{z} = \mathbf{0} \tag{8.73}$$

$$g_i(\mathbf{x}) = 0 \qquad i = 1,\ldots,m \tag{8.74}$$

$$x_i z_i - \mu = 0 \qquad i = 1,\ldots,n \tag{8.75}$$

If we define **e** as the vector of 1's of *n* dimension and,

$$\mathbf{X} = \begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{pmatrix}$$

we can replace (8.75) above with (8.78) below,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) - \mathbf{z} = \mathbf{0} \tag{8.76}$$

$$g_i(\mathbf{x}) = 0 \quad i = 1,\ldots,m \tag{8.77}$$

$$\mathbf{XZe} - \mu\mathbf{e} = 0 \tag{8.78}$$

As $\mu \to 0$, the above equations (along with $\mathbf{z} \geq 0$) represent the KKT conditions for the original problem. The variables **z** can be viewed as the Lagrange multipliers for the bound constraints. Notice that with $\mu = 0$, (8.78) expresses complementary slackness for the bound constraints, i.e. either $x_i = 0$ or $z_i = 0$.

8.4.7 Problem Solution

We will now construct the coefficient matrix for the Newton iteration. We note that we have *n* equations from (8.73), *m* equations from (8.77) and *n* equations from (8.75). Similarly, these equations are functions of *n* variables *x*, *m* variables λ, and *n* variables *z*.

For example, the first row of the coefficient matrix for NR could be represented as,

$$\begin{bmatrix} \left(\nabla_x f_{1NR}\right)^T & \left(\nabla_\lambda f_{1NR}\right)^T & \left(\nabla_z f_{1NR}\right)^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\lambda \\ \Delta\mathbf{z} \end{bmatrix} = -\begin{bmatrix} r1 \end{bmatrix} \tag{8.79}$$

where f_{1NR} is given by,

$$f_{1NR} = \frac{\partial f}{\partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_1} - z_1 \tag{8.80}$$

and *r1* represents the residuals for (8.76). If we substitute (8.80) into matrix (8.79), the first row of the coefficient matrix is,

$$\underbrace{\left[\frac{\partial^2 f}{\partial x_1^2} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_1^2} - \frac{\partial z_1}{\partial x_1}\right], \ldots, \left[\frac{\partial^2 f}{\partial x_n \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_n \partial x_1} - \frac{\partial z_1}{\partial x_n}\right]}_{\nabla_x f^T}, \underbrace{\left[-\frac{\partial g_1}{\partial x_1}\right], \ldots, \left[-\frac{\partial g_m}{\partial x_1}\right]}_{\nabla_\lambda f^T}, \underbrace{[-1], \ldots, [0]}_{\nabla_z f^T}$$

Defining,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i + \mu \mathbf{X}^{-2} \tag{8.81}$$

we can write the NR iteration equations at step *k* as,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k & \left(-\mathbf{J}^k\right)^T & -\mathbf{I} \\ \mathbf{J}^k & 0 & 0 \\ \mathbf{Z}^k & 0 & \mathbf{X}^k \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\lambda^k \\ \Delta\mathbf{z}^k \end{bmatrix} = -\begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} \tag{8.82}$$

The reader might wish to compare this with (8.56). As before, we could solve this system of equations to obtain $\Delta\mathbf{x}^k$, $\Delta\boldsymbol{\lambda}^k$, and $\Delta\mathbf{z}^k$. However, we can simplify this expression. We will start by looking at the third set of equations in (8.56) above,

$$\mathbf{Z}^k \Delta\mathbf{x}^k + 0 + \mathbf{X}^k \Delta\mathbf{z}^k = -\mathbf{X}^k \mathbf{Z}^k \mathbf{e} + \mu\mathbf{e} \tag{8.83}$$

where we have substituted in the actual residual value for *r3*, i.e.,

$$-r3 = -\mathbf{X}^k \mathbf{Z}^k \mathbf{e} + \mu\mathbf{e}$$

We would like to solve (8.83) for $\Delta\mathbf{z}^k$. Rearranging terms gives,

$$\mathbf{X}^k \Delta \mathbf{z}^k = -\mathbf{X}^k \mathbf{Z}^k \mathbf{e} + \mu \mathbf{e} - \mathbf{Z}^k \Delta \mathbf{x}^k \tag{8.84}$$

If we pre-multiply both sides by $(\mathbf{X}^k)^{-1}$ we have,

$$\Delta \mathbf{z}^k = -\mathbf{Z}^k \mathbf{e} + \mu (\mathbf{X}^k)^{-1} \mathbf{e} - (\mathbf{X}^k)^{-1} \mathbf{Z}^k \Delta \mathbf{x}^k \tag{8.85}$$

Examining the first and second terms on the right hand side, we note that,

$$-\mathbf{Z}^k \mathbf{e} = -\begin{pmatrix} z_1^k & & 0 \\ & \ddots & \\ 0 & & z_n^k \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -\mathbf{z}^k \qquad \mu (\mathbf{X}^k)^{-1} \mathbf{e} = \begin{pmatrix} \frac{\mu}{x_1^k} & & 0 \\ & \ddots & \\ 0 & & \frac{\mu}{x_n^k} \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{z}^k$$

Equation (8.85) becomes,

$$\begin{aligned} \Delta \mathbf{z}^k &= -\mathbf{z}^k + \mathbf{z}^k - (\mathbf{X}^k)^{-1} \mathbf{Z}^k \Delta \mathbf{x}^k \\ &= -(\mathbf{X}^k)^{-1} \mathbf{Z}^k \Delta \mathbf{x}^k \end{aligned} \tag{8.86}$$

We define **S** to be,

$$\mathbf{S}^k = (\mathbf{X}^k)^{-1} \mathbf{Z}^k \tag{8.87}$$

so that we have,

$$\Delta \mathbf{z}^k = -\mathbf{S}^k \Delta \mathbf{x}^k \tag{8.88}$$

Now we will examine the first two rows of (8.82). These represent equations,

$$\nabla_x^2 L^k \Delta \mathbf{x}^k + \left(-\mathbf{J}^k\right)^T \Delta \lambda^k + -\mathbf{I} \Delta \mathbf{z}^k = -r1 \tag{8.89}$$

$$\mathbf{J}^k \Delta \mathbf{x}^k = -r2 \tag{8.90}$$

We see that $\Delta \mathbf{z}^k$ only appears in (8.89). If we substitute (8.88) for $\Delta \mathbf{z}^k$ and gather terms, (8.89) becomes,

$$\left[\nabla_x^2 L^k + \mathbf{S}^k\right] \Delta \mathbf{x}^k + \left(-\mathbf{J}^k\right)^T \Delta \lambda^k = -r1 \tag{8.91}$$

or in matrix form,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k + \mathbf{S} & \left(-\mathbf{J}^k\right)^T \\ \mathbf{J}^k & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \lambda^k \end{bmatrix} = -\begin{bmatrix} r1 \\ r2 \end{bmatrix} \tag{8.92}$$

Once we have solved for $\Delta \mathbf{x}^k$, we can use (8.88) to get $\Delta \mathbf{z}^k$.

8.4.8 Example 1: Solving the IP Equations

We will illustrate this approach on a similar example problem used earlier:

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 \geq 0 \\ & x_1 \geq 0 \end{aligned}$$

We will change the inequality constraint to an equality constraint by means of a slack variable, x_2, so that we have,

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 - x_2 = 0 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We will eliminate the lower bounds by using a barrier formulation:

$$\begin{aligned} \text{Min} \quad & f_\mu = x_1^2 - \mu \sum_{i=1}^{2} \ln(x_i) \\ \text{s.t.} \quad & -2x_1 - x_2 + 9 = 0 \end{aligned}$$

starting from $(\mathbf{x}^0)^T = [3,3]$. At this point, $f^0 = 9, (\nabla f^0)^T = [6,0]$; $g^0 = 0, (\nabla g^0)^T = [-2,-1]$.

We will set $\mu = 10$ and $\lambda = 1$. This then gives, $z_1 = \dfrac{\mu}{x_1} = 3.333$ and $z_2 = \dfrac{\mu}{x_2} = 3.333$ Noting,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i + \mu \mathbf{X}^{-2} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} - (1) \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1.111 & 0 \\ 0 & 1.111 \end{bmatrix}$$

$$\mathbf{J}^k = [-2, -1]$$

We can write the coefficient matrix,

$$\begin{bmatrix} \nabla_x^2 L^k & \left(-\mathbf{J}^k\right)^T & -\mathbf{I} \\ \mathbf{J}^k & 0 & 0 \\ \mathbf{Z}^k & 0 & \mathbf{X}^k \end{bmatrix} = \begin{bmatrix} 3.111 & 0 & 2 & -1 & 0 \\ 0 & 1.111 & 1 & 0 & -1 \\ -2 & -1 & 0 & 0 & 0 \\ 3.333 & 0 & 0 & 3 & 0 \\ 0 & 3.333 & 0 & 0 & 3 \end{bmatrix}$$

By evaluating (8.76) through (8.78) we find the vector of residuals to be,

$$-\begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} = -\begin{bmatrix} 4.667 \\ -2.333 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving this set of equations, gives $\Delta\mathbf{x} = \begin{bmatrix} -0.712 \\ 1.424 \end{bmatrix}$, $\Delta\lambda = -0.831$, $\Delta\mathbf{z} = \begin{bmatrix} 0.791 \\ -1.582 \end{bmatrix}$

If we take the full step by adding these delta values to our beginning values, we have

$$\mathbf{x}^1 = \begin{bmatrix} 2.288 \\ 4.424 \end{bmatrix}, \quad \lambda^1 = 0.169, \quad \mathbf{z}^1 = \begin{bmatrix} 4.124 \\ 1.805 \end{bmatrix}$$

At this new point the constraint is satisfied ($g^1 = 0$) and the objective has decreased from 9 to 5.235.

8.5 The Generalized Reduced Gradient (GRG) Algorithm

8.5.1 Introduction

GRG works quite differently than the SQP or IP methods. If started inside feasible space, GRG goes downhill until it runs into fences—constraints—and then corrects the search direction such that it follows the fences downhill. At every step it enforces feasibility. The strategy of GRG in following fences works well for engineering problems because most engineering optimums are constrained. For information beyond what is given here consult Lasdon et al. [30] and Gabriele and Ragsdell [31].

8.5.2 Explicit vs. Implicit Elimination

Suppose we have the following optimization problem,

$$\text{Min} \quad f(\mathbf{x}) = x_1^2 + 3x_2^2 \tag{8.93}$$

$$\text{s.t.} \qquad g(\mathbf{x}) = 2x_1 + x_2 - 6 = 0 \tag{8.94}$$

A contour plot is given in Fig. 8.9a.

From previous discussions about modeling in Chapter 2, we know there are two approaches to this problem—we can solve it as a problem in two variables with one equality constraint, or we can use the equality constraint to eliminate a variable and the constraint. We will use the second approach. Using (8.94) to solve for x_2,

$$x_2 = 6 - 2x_1$$

Substituting into the objective function, (8.93), we have,

$$\text{Min} \qquad f(\mathbf{x}) = x_1^2 + 3(6 - 2x_1)^2 \tag{8.95}$$

Mathematically, solving the problem given by (8.93)-(8.94) is the same as solving the problem in (8.95). We have used the constraint to *explicitly* eliminate a variable and a constraint. Once we solve for the optimal value of x_1, we will obviously have to back substitute to get the value of x_2 using (8.94). The solution in x_1 is illustrated in Fig. 8.9b, where the sensitivity plot for (8.95) is given (because we only have one variable, we can't show a contour plot). The derivative $\dfrac{df}{dx_1}$ of (8.95) would be considered to be the *reduced gradient* relative to the original problem.

Usually we cannot make an explicit substitution as we did in this example. So we eliminate variables *implicitly*. We show how this can be done in the next section.

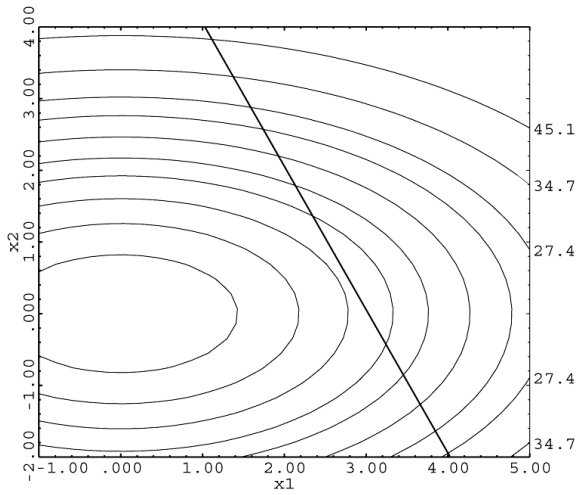

Fig. 8.9 a) Contour plot in x_1, x_2 with equality constraint. The optimum is at $\mathbf{x}^T = \begin{bmatrix} 2.7693 & 0.4613 \end{bmatrix}$.

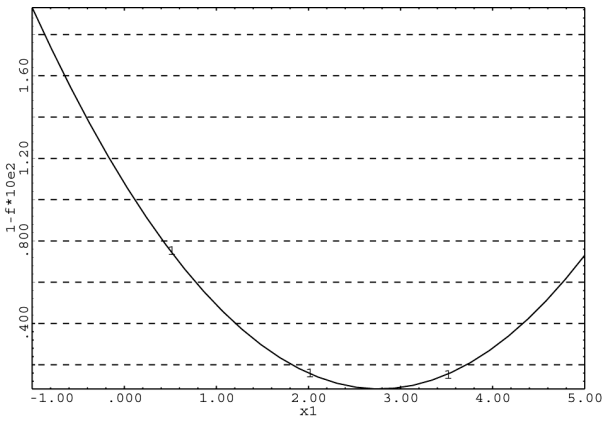

Fig. 8.9 b) Sensitivity plot for Eq. 8.95. The optimum is at $x_1 = 2.7693$

8.5.3 Implicit Elimination

In this section we will look at how we can eliminate variables implicitly. We do this by considering *differential* changes in the objective and constraints. We will start by considering a simple problem of two variables with one equality constraint,

$$\begin{aligned} &\text{Min} \quad f(\mathbf{x}) \quad \mathbf{x}^T = [x_1 \quad x_2] \\ &\text{s.t.} \quad g(\mathbf{x}) = 0 \end{aligned}$$

Suppose we are at a feasible point. Thus the equality constraint is satisfied. We wish to move to improve the objective function. The differential change is given by,

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 \tag{8.96}$$

to keep the constraint satisfied the differential change must be zero:

$$dg = \frac{\partial g}{\partial x_1} dx_1 + \frac{\partial g}{\partial x_2} dx_2 = 0 \tag{8.97}$$

Solving for dx_2 in (7.47) gives:

$$dx_2 = \frac{-\partial g / \partial x_1}{\partial g / \partial x_2} dx_1$$

substituting into (7.46) gives,

$$df = \left[\frac{\partial f}{\partial x_1} - \frac{\partial f}{\partial x_2} \left(\frac{\partial g / \partial x_1}{\partial g / \partial x_2} \right) \right] dx_1 \tag{8.98}$$

where the term in brackets is the reduced gradient.

i.e., $$\frac{df_R}{dx_1} = \left[\frac{\partial f}{\partial x_1} - \frac{\partial f}{\partial x_2} \left(\frac{\partial g / \partial x_1}{\partial g / \partial x_2} \right) \right] \tag{8.99}$$

If we substitute Δx for dx, then the equations are only approximate. *We are stepping tangent to the constraint in a direction that improves the objective function.*

8.5.4 GRG Algorithm with Equality Constraints Only

We can extend the concepts of the previous section to the general problem which we represent in vector notation. Suppose now we consider the general problem with equality constraints,

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) = 0 \quad i = 1,\ldots,m \end{aligned}$$

We have *n* design variables and *m* equality constraints. We begin by partitioning the design variables into (*n*-*m*) independent variables, **z**, and *m* dependent variables **y**. The independent variables will be used to improve the objective function, and the dependent variables will be used to satisfy the binding constraints. If we partition the gradient vectors as well we have,

$$\nabla f(\mathbf{z})^{\mathrm{T}} = \left[\begin{array}{cccc} \frac{\partial f(\mathbf{x})}{\partial z_1} & \frac{\partial f(\mathbf{x})}{\partial z_2} & \cdots & \frac{\partial f(\mathbf{x})}{\partial z_{n-m}} \end{array} \right]$$

$$\nabla f(\mathbf{y})^{\mathrm{T}} = \left[\begin{array}{cccc} \frac{\partial f(\mathbf{x})}{\partial y_1} & \frac{\partial f(\mathbf{x})}{\partial y_1} & \cdots & \frac{\partial f(\mathbf{x})}{\partial y_m} \end{array} \right]$$

We will also define independent and dependent matrices of the partial derivatives of the constraints:

$$\frac{\partial \psi}{\partial \mathbf{z}} = \left[\begin{array}{cccc} \frac{\partial g_1}{\partial z_1} & \frac{\partial g_1}{\partial z_2} & \cdots & \frac{\partial g_1}{\partial z_{n-m}} \\ \frac{\partial g_m}{\partial z_1} & \frac{\partial g_m}{\partial z_2} & \cdots & \frac{\partial g_m}{\partial z_{n-m}} \end{array} \right] \qquad \frac{\partial \psi}{\partial \mathbf{y}} = \left[\begin{array}{cccc} \frac{\partial g_1}{\partial y_1} & \frac{\partial g_1}{\partial y_2} & \cdots & \frac{\partial g_m}{\partial y_m} \\ \frac{\partial g_m}{\partial y_1} & \frac{\partial g_m}{\partial y_2} & \cdots & \frac{\partial g_m}{\partial y_m} \end{array} \right]$$

We can write the differential changes in the objective and constraints in vector form as:

$$df = \nabla f(\mathbf{z})^{\mathrm{T}} d\mathbf{z} + \nabla f(\mathbf{y})^{\mathrm{T}} d\mathbf{y} \tag{8.100}$$

$$d\psi = \frac{\partial \psi}{\partial \mathbf{z}} d\mathbf{z} + \frac{\partial \psi}{\partial \mathbf{y}} d\mathbf{y} = \mathbf{0} \tag{8.101}$$

Noting that $\frac{\partial \psi}{\partial \mathbf{y}}$ is a square matrix, and solving (8.101) for d**y**,

$$d\mathbf{y} = -\frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} d\mathbf{z} \tag{8.102}$$

substituting (8.102) into (8.100),

$$df = \nabla f(\mathbf{z})^{\mathrm{T}} d\mathbf{z} - \nabla f(\mathbf{y})^{\mathrm{T}} \frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} d\mathbf{z}$$

or $$\nabla f_R^{\mathrm{T}} = \nabla f(\mathbf{z})^{\mathrm{T}} - \nabla f(\mathbf{y})^{\mathrm{T}} \frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} \tag{8.103}$$

where ∇f_R^{T} is the reduced gradient. *The reduced gradient is the direction of steepest ascent that stays tangent to the binding constraints*.

8.5.5 GRG Example 1: One Equality Constraint

We will illustrate the theory of the previous section with the following example. For this example we will have three variables and one equality constraint. We state the problem as,

$$\text{Min} \quad f = 4x_1^2 + x_2^2 + 3x_3^2$$

$$\text{s.t.} \quad g = 2x_1 + 4x_2 - x_3 = 10$$

Step 1: Evaluate the objective and constraints at the starting point.
The starting point will be $\mathbf{x}^T = [2 \quad 2 \quad 2]$, at which point $f = 32$ and $g = 10$. So the constraint is satisfied.

Step 2: Partition the variables.
We have one binding constraint so we will need one dependent variable. We will arbitrarily choose x_1 as the dependent variable, so $\mathbf{y} = [x_1]$. The independent variables will therefore be $\mathbf{z}^T = [x_2 \quad x_3]$. Thus,

$$\mathbf{z} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \quad \mathbf{y} = [x_1] \quad \nabla f(\mathbf{z}) = \begin{bmatrix} \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 2x_2 \\ 6x_3 \end{bmatrix}_{@2,2} = \begin{bmatrix} 4 \\ 12 \end{bmatrix} \quad \nabla f(\mathbf{y}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \end{bmatrix} = [8x_1]_{@2} = [16]$$

$$\frac{\partial \psi}{\partial z} = \begin{bmatrix} \frac{\partial g}{\partial x_2} & \frac{\partial g}{\partial x_3} \end{bmatrix} = [4 \quad -1] \qquad \frac{\partial \psi}{\partial y} = \begin{bmatrix} \frac{\partial g}{\partial x_1} \end{bmatrix} = [2]$$

Step 3: Compute the reduced gradient.
We now have the information we need to compute the reduced gradient:

$$\nabla f_R^{\mathrm{T}} = \nabla f(\mathbf{z})^{\mathrm{T}} - \nabla f(\mathbf{y})^{\mathrm{T}} \frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}}$$

$$\nabla f_R^{\mathrm{T}} = [4 \quad 12] - [16] \left[\frac{1}{2} \right] [4 \quad -1]$$

$$= [-28 \quad 20]$$

Step 4: Compute the direction of search.

We will step in the direction of steepest descent, i.e., the negative reduced gradient direction, which is the direction of steepest descent which stays tangent to the constraint.

$$\mathbf{s} = \begin{bmatrix} 28 \\ -20 \end{bmatrix} \qquad \text{or, normalized, } \mathbf{s} = \begin{bmatrix} 0.8137 \\ -0.5812 \end{bmatrix}$$

Step 5: Do a line search in the independent variables
We will use our regular formula,

$$\mathbf{z}^{new} = \mathbf{z}^{old} + \alpha \mathbf{s}$$

We will arbitrarily pick a starting step length $\alpha = 0.5$

$$\begin{bmatrix} x_2^{new} \\ x_3^{new} \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} + 0.5 \begin{bmatrix} 0.8137 \\ -0.5812 \end{bmatrix} = \begin{bmatrix} 2.4068 \\ 1.7094 \end{bmatrix}$$

Step 6: Solve for the value of the dependent variable.
We do this using (7.52) above, only we will substitute Δy for dy:

$$\Delta \mathbf{y} = -\frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} \Delta \mathbf{z}$$

$$\begin{aligned} [\Delta x_1] &= -\frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} \begin{bmatrix} \Delta x_2 \\ \Delta x_3 \end{bmatrix} \\ &= -\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 4 & -1 \end{bmatrix} \begin{bmatrix} 0.4069 \\ -0.2906 \end{bmatrix} \\ &= -0.9590 \end{aligned}$$

So the new value of x_1 is,

$$\begin{aligned} x_1^{new} &= x_1^{old} + \Delta x \\ &= 2 - 0.9590 \\ &= 1.041 \end{aligned}$$

Our new point is $\mathbf{x}^T = \begin{bmatrix} 1.041 & 2.4069 & 1.7094 \end{bmatrix}$ at which point $f = 18.9$ and $g = 10$. We observe that the objective has decreased from 32 to 18.9 and the constraint is still satisfied. This only represents one step in the line search. We would continue the line search until we reach a minimum.

8.5.6 GRG Algorithm with Equality and Inequality Constraints

In this section we will consider the general problem with both inequality and equality constraints,

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \le 0 \quad i = 1,\ldots,k \\ & g_i(\mathbf{x}) = 0 \quad i = k+1,\ldots,m \end{aligned}$$

The extension of the GRG algorithm to include inequality constraints involves some additional complexity, because the derivation of GRG is based on equality constraints. We therefore convert inequalities into equalities by adding slack variables.

The GRG algorithm described here is an *active constraint* algorithm—only the binding inequality constraints are used to determine the search direction. The non-binding constraints enter into the problem only if they become binding or violated.

8.5.7 Steps of the GRG Algorithm for the General Problem

1. Evaluate the objective function and all constraints at the current point.

2. For any binding inequality constraints, add a slack variable, s_i

3. Partition the variables into independent variables and dependent variables. We will need one dependent variable for each binding constraint. Any variable at either its upper or lower limit should become an independent variable.

4. Compute the reduced gradient using (8.103).

5. Calculate a direction of search. We can use any method to calculate the search direction that relies on gradients since the reduced gradient is a gradient. For example, we can use a quasi-Newton update.

6. Do a line search in the independent variables. For each step, find the corresponding values in the dependent variables using (8.102) with $\Delta\mathbf{z}$ and $\Delta\mathbf{y}$ substituted for $d\mathbf{z}$ and $d\mathbf{y}$.

7. At each step in the line search, drive back to the constraint boundaries for any violated constraints using Newton-Raphson to adjust the dependent variables. If an independent variable hits its bound, set it equal to its bound.

   The NR iteration is given by $\Delta\mathbf{y} = -\dfrac{\partial\psi^{-1}}{\partial\mathbf{y}}(\mathbf{g}-\mathbf{b})$ We note we already have the matrix $\dfrac{\partial\psi^{-1}}{\partial\mathbf{y}}$ from the calculation of the reduced gradient.

8. The line search may terminate either of 4 ways

   1) The minimum in the direction of search is found (using, for example, quadratic interpolation).

2) A dependent variable hits its upper or lower limit.

3) A formerly non-binding constraint becomes binding.

4) NR fails to converge. In this case we must cut back the step size until NR does converge.

9. If at any point the reduced gradient in step 4 is equal to **0**, the KKT conditions are satisfied.

8.5.8 GRG Example 2: Two Inequality Constraints

In this problem we have two inequality constraints and will therefore need to add in slack variables.

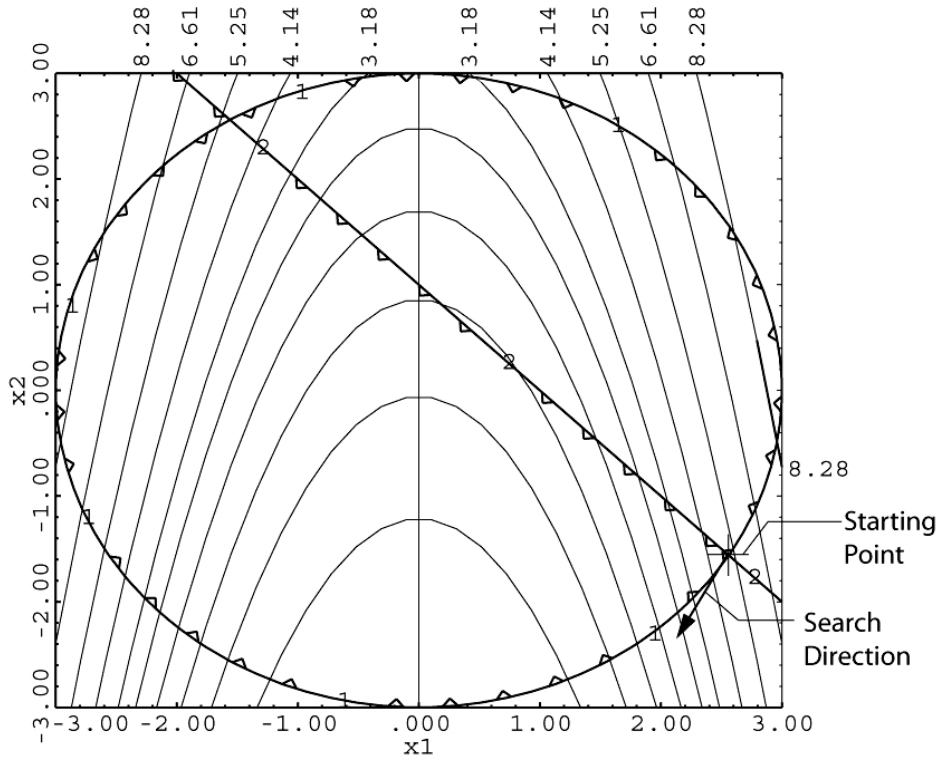

Fig. 8.10 Example problem for GRG algorithm

$$\text{Min.} \quad f(\mathbf{x}) = x_1^2 + x_2$$

$$\text{s.t.:} \quad g_1(\mathbf{x}) = x_1^2 + x_2^2 - 9 \le 0$$

$$g_2(\mathbf{x}) = x_1 + x_2 - 1 \le 0$$

Suppose, to make things interesting, we are starting at $\mathbf{x}^T = [2.56155, -1.56155]$ where both constraints are binding.

Step 1: Evaluate functions.

$$f(\mathbf{x}) = 5.0 \qquad g_1(\mathbf{x}) = 0.0 \qquad g_2(\mathbf{x}) = 0.0$$

Step 2: Add in slack variables.

We note that both constraints are binding so we will add in two slack variables. $s_1,\ s_2$.

Step 3: Partition the variables

Since the slack variables are at their lower limits (=0) they will become the independent variables; x_1, x_2 will be the dependent variables.

$$\mathbf{z}^T = \begin{bmatrix} s_1 & s_2 \end{bmatrix} \qquad \mathbf{y}^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

Step 4: Compute the reduced gradient

$$\nabla f(\mathbf{z})^T = \begin{bmatrix} 0.0 & 0.0 \end{bmatrix} \qquad \nabla f(\mathbf{y})^T = \begin{bmatrix} 5.123 & 1.0 \end{bmatrix}$$

$$\frac{\partial \psi}{\partial \mathbf{z}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \frac{\partial \psi}{\partial \mathbf{y}} = \begin{bmatrix} 5.123 & -3.123 \\ 1.0 & 1.0 \end{bmatrix}$$

$$\frac{\partial \psi^{-1}}{\partial \mathbf{y}} = \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix}$$

thus

$$\frac{\partial \psi^{-1}}{\partial \mathbf{y}} \frac{\partial \psi}{\partial \mathbf{z}} = \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix}$$

$$\begin{aligned} \nabla f_r^T &= \begin{bmatrix} 0.0 & 0.0 \end{bmatrix} - \begin{bmatrix} 5.123 & 1 \end{bmatrix} \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix} \\ &= \begin{bmatrix} 0.0 & 0.0 \end{bmatrix} - \begin{bmatrix} 0.50 & 2.56 \end{bmatrix} \\ &= \begin{bmatrix} -0.50 & -2.56 \end{bmatrix} \end{aligned}$$

Step 5: Calculate a search direction.

We want to move in the negative gradient direction, so our search direction will be $\mathbf{s}^T = \begin{bmatrix} 0.50 & 2.56 \end{bmatrix}$. This is the direction for the independent variables (the slacks). When normalized this direction is $\mathbf{s}^T = \begin{bmatrix} 0.19 & 0.98 \end{bmatrix}$.

Step 6: Conduct the line search in the independent variables

We will start our line search, denoting the current point as $\mathbf{z}^0$,

$$\mathbf{z}^1 = \mathbf{z}^0 + \alpha \mathbf{s}^0$$

Suppose we pick $\alpha = 1.0$. Then

$$\mathbf{z}^1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (1.0)\begin{bmatrix} 0.19 \\ 0.98 \end{bmatrix}$$
$$\mathbf{z}^1 = \begin{bmatrix} 0.19 \\ 0.98 \end{bmatrix}$$

Step 7: Adjust the dependent variables
To find the change in the dependent variables, we use (7.52)

$$\Delta\mathbf{y} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = -\left[\frac{\partial \psi^{-1}}{\partial \mathbf{y}}\right]\left[\frac{\partial \psi}{\partial \mathbf{z}}\right]\Delta\mathbf{z}$$
$$= \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix}\begin{bmatrix} 0.19 \\ 0.98 \end{bmatrix}$$
$$= \begin{bmatrix} -0.394 \\ -0.586 \end{bmatrix}$$

$$x_1^1 = 2.56155 - 0.394 = 2.168$$
$$x_2^1 = -1.56155 - 0.586 = -2.148$$

at which point $f(\mathbf{x}) = 2.522$

Have we violated any constraints?

$$g_1(\mathbf{x}) = x_1^2 + x_2^2 - 9 = (2.168)^2 + (-2.148)^2 - 9 = 0.31 \text{ (violated)}$$
$$g_2(\mathbf{x}) = x_1 + x_2 - 1 = 2.168 - 2.148 - 1 = -0.98 \text{ (satisfied)}$$

We need to drive back to where the violated constraint is satisfied. We will use NR to do this. Since we don't want to drive back to where both constraints are binding, we will set the residual for constraint 2 to zero.

NR Iteration 1:

$$\mathbf{y}^{(n)} = \mathbf{y}^{(0)} - \frac{\partial \psi^{-1}}{\partial \mathbf{y}}(\mathbf{g} - \mathbf{b})$$
$$= \begin{bmatrix} 2.168 \\ -2.148 \end{bmatrix} - \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix}\begin{bmatrix} 0.31 \\ 0.0 \end{bmatrix}$$
$$= \begin{bmatrix} 2.130 \\ -2.110 \end{bmatrix}$$

at this point

$$g_1 = (2.130)^2 + (2.110)^2 - 9 = -0.011$$
$$g_2 = -0.98$$

NR Iteration 2:

$$= \begin{bmatrix} 2.130 \\ -2.110 \end{bmatrix} - \begin{bmatrix} 0.1213 & 0.3787 \\ -0.1213 & 0.6213 \end{bmatrix} \begin{bmatrix} -0.011 \\ 0.0 \end{bmatrix}$$
$$= \begin{bmatrix} 2.1313 \\ -2.113 \end{bmatrix}$$

evaluating constraints:

$$g_1 = (2.1313)^2 + (-2.113)^2 - 9 = 0$$
$$g_2 = -0.98$$

We are now feasible again. We have taken one step in the line search!

Our new point is $\mathbf{x} = \begin{bmatrix} 2.1313 \\ -2.113 \end{bmatrix}$ at which point the objective is 2.43, and all constraints are satisfied.

We would continue the line search until we run into a new constraint boundary, a dependent variable hits a bound, or we can no longer get back on the constraint boundaries (which is not an issue in this example, since the constraint is linear).